

ISBN 번호

과학데이터 관리 및 유통을 위한
메타데이터 가이드

ver.1.0

한국과학기술정보연구원

과학데이터 관리 및 유통을 위한
메타데이터 표준 가이드

ver.1.0

한국과학기술정보연구원

지은이

김선태 / KISTI

박형선 / KISTI

이상환 / KISTI

이상호 / KISTI

이승복 / KISTI

이정훈 / KISTI

최명석 / KISTI

곽남영 / 투이컨설팅

< 목 차 >

1. 메타데이터란 무엇인가?	1
1.1 목적과 범위	1
1.2 메타데이터 정의	1
1.3 메타데이터의 필요성	2
1.4 메타데이터 요소의 유형	3
1.5 메타데이터 스키마	4
1.6 메타데이터 표준의 필요성	5
1.7 메타데이터 요소의 사용과 이름공간(namespace)	6
2. 과학데이터 유통을 위한 메타데이터	10
2.1 목적과 범위	10
2.2 과학데이터 유통 구조	10
2.3 과학데이터 구조 별 정의사례	11
2.4 컬렉션 메타데이터	13
2.5 아이템 메타데이터	18
2.6 파일 메타데이터	28
3. 리파지토리 관리를 위한 메타데이터	44
3.1 목적과 범위	44
3.2 리파지토리 메타데이터 스키마	44
3.3 요소 정의 및 작성 방법	46
3.4 XML 작성 예시	54
4. 메타데이터 전송	55
4.1 목적과 범위	55
4.2 직접 전송	56
4.3 OAI-PMH 프로토콜을 이용한 전송	57
4.4 metadataPrefix and Metadata Schema	72
4.5 요청과 응답의 흐름 제어 (flow control)	73
4.6 프로토콜 요청과 응답	75
5. 참고자료	95

1. 메타데이터란 무엇인가?

1.1 목적과 범위

이 문서는 연구자나, 연구자를 지원하는 업무 담당자, 도서관이나 데이터센터 직원, 리포지터리(repository) 운영자 등에게 메타데이터에 대한 개념을 전달 하는데 있다. 메타데이터는 그 용도에 따라 종류 또한 다양하다. 예를 들어, 컴퓨터 프로그램이 작동하기 위해 필요한 여러 가지 정보를 XML로 작성하여 시스템적으로 사용되는 메타데이터가 그 중 하나이다. 하지만 이 가이드에서는 이러한 메타데이터는 제외하고 정보자원을 수집, 저장, 관리, 보존, 배포(출판) 과 관련된 메타데이터만을 설명한다.

1.2 메타데이터 정의

본 문서에서 데이터는 오프라인과 온라인상에서 접근 가능한 모든 정보자원 (information resources, item, object)을 의미한다. 정보자원은 도서와 같은 물리적인 자원뿐만 아니라 컴퓨터 파일, 문서, 이미지, 데이터세트와 같은 디지털 자원을 모두 의미한다. 메타데이터는 ‘정보자원에 대한 데이터’이다. ‘데이터에 대한 데이터’가 메타데이터의 보편적 정의로 사용되는 이유이다. 이상을 종합해 볼 때 메타데이터란 정보자원이나 정보자원의 묶음(collection, object)을 설명(기술, describe)한 데이터를 말한다.

메타데이터의 대표적인 예로 도서관의 목록(catalog)을 들 수 있다. 도서관 목록은 도서나 저널, 그 밖에 도서관이 보유(소장)하고 있는 정보자원에 대한 정보로 구성되어 있다. 예를 들어, 어떤 도서(책)의 제목이나 저자, 출판사, 출판년도 등의 정보가 해당 도서(정보자원)에 대한 정보로서 목록의 내용을 구성하고 있다. 이렇듯 책이라는 정보자원을 설명하는 제목, 저자, 출판사 등을 메타데이터 요소(metadata elements, property, 속성)라 하며, 그 값을 메타데이터 요소의 속성값(속성치)이라 한다.

사진이라는 정보자원(아이템, 데이터라고도 함)을 하나의 예로 더 살펴보면, 사진을 찍은 사람, 사진 속 인물은 누구인지, 사진을 언제 찍었는지, 어디에서 찍었는지, 어떤 카메라로 찍었는지 등의 정보는 사진이라는 정보자원에 대한



그림 1 디지털 카메라의 메타데이터 설정
 사진출처: <http://bit.ly/1JD2SJJ>

데이터(메타데이터)이다. 이 가이드의 ‘메타데이터의 종류’ 부분에서 자세히 언급 되겠지만, 메타데이터는 사람이 직접 입력해야 하는 것도 있고, 정보자원을 생산하는 생산장비나 정보자원을 관리하는 시스템 등에서 자동적으로 생산되는 것도 있다. 사진의 경우, ‘사진 속 인물이 누구인지’는 사람이 직접 만들어야하는 메타데이터 이지만, ‘사진을 찍은 사람’과 ‘사진을 언제 찍었는지’ 등의 정보는 카메라

에서 자동으로 생산해 낼 수 있는 메타데이터이다. 사진의 해상도나 사진이 찍힌 위치정보(좌표값) 등도 후자에 해당된다.

정보자원의 묶음(collection)이란 여러 개의 정보자원을 하나로 모은 것을 말한다. 예를 들어, 디지털 사진(정보자원, object, item, resource)을 여러 장을 CD(compact disc)에 저장했을 때, 이 한 장의 CD가 컬렉션이 될 수 있으며, 인화된 사진을 모아놓은 앨범도 컬렉션이 될 수 있다. 물론 디지털 사진을 모아 놓은 하나의 컴퓨터 폴더(folder)도 컬렉션이 될 수 있다. 폴더가 폴더를 포함할 수 있는 것과 같이 컬렉션은 컬렉션을 포함할 수 있다.

이러한 컬렉션(정보자원의 묶음)을 설명하기 위해서도 메타데이터가 필요하다. 예를 들어, 컬렉션 이름, 컬렉션의 크기(size), 누가 컬렉션을 만들었는지? 언제 만들었는지? 등의 정보(메타데이터)는 컬렉션 자체가 아니라 컬렉션을 설명(기술)하는 데이터이다. 이러한 데이터가 컬렉션을 설명하는 메타데이터이다.

하나의 사진파일도 ‘사진 이름(title)’이 있을 수 있으며, 폴더나 CD도 ‘폴더 이름(title)’이나 ‘CD 이름(title)’이 있을 수 있다. ‘title’이라는 동일한 메타데이터 속성으로 각각의 속성치(속성값)을 기술(describe) 할 수 있다.

1.3 메타데이터의 필요성

정보자원에 따라서 메타데이터의 중요성에는 다소 차이가 있다. 하지만 메타데이터의 공통적 기능은 정보자원의 유형과 관계없이 거의 동일하다. 이는 메타데이터의 고유기능이 존재함을 의미한다. 메타데이터가 사용되는 용도에 따라 대표적인 메타데이터 스키마가 존재하는 것은 메타데이터의 고유기능이 세분

화 될 수 있음을 의미한다.

메타데이터는 정보자원을 설명(기술)한 데이터이다. 따라서 메타데이터를 통해서 데이터가 존재하게 된 관련정보(context information, 상황정보)를 알 수 있다. 따라서 사진이나, 문서, 동영상, 도서와 같은 정보자원과 달리, 단순한 수치 데이터만을 담고 있는 정보자원의 경우 메타데이터의 효용성이 더욱 크다. 예를 들어, 특정 실험 장치로 데이터(정보자원, 원시데이터)를 획득했을 때, 실험의 조건이나 장치이름, 장치버전, 장치에 탑재된 소프트웨어 이름과 버전 등의 정보는 데이터가 생산되는 시점의 상황 정보를 알 수 있다. 따라서 데이터를 이해하는데 정보를 제공할 수 있으며, 데이터의 재사용을 보장할 수 있다.

1.4 메타데이터 요소의 유형

메타데이터 요소들의 쓰임새에 따른 유형별 구분은 다음과 같다. 사진(정보자원)을 가지고 사진과 관련된 메타데이터 요소를 살펴보면 다음과 같다.

- Descriptive Metadata
 - (사진) 데이터를 설명하기 위한 메타데이터
 - 메타데이터 예: 사진을 찍은 사람(photographer), 사진의 주제(subject), 사진을 찍은 일시(date, time)

- Technical Metadata
 - (사진) 데이터의 기술적 특징을 담기 위한 메타데이터
 - 메타데이터 예: 카메라 유형(type), 사진 크기(dimensions), 해상도(resolution)

- Access or rights metadata
 - (사진) 데이터의 접근과 이용(라이선스) 정보를 담기 위한 메타데이터
 - 메타데이터 예: 접근 범위(access), 접근 조건(right, license)

- Preservation metadata
 - (사진) 데이터의 지속적 접근과 활용을 보장하기 위한 메타데이터
 - 메타데이터 예: 데이터 변환일(date), 데이터 버전(version), 운영시스템 종류(os)

1.5 메타데이터 스키마

상기의 메타데이터 요소들(elements, terms)을 조합하여 특정한 목적의 메타데이터 요소세트를 구성한 것이 메타데이터 스키마이다. 메타데이터 스키마는 정보자원을 위한 메타데이터 요소들의 구성 순서와 반복사용 여부 및 횟수, 메타데이터 요소 사용에 있어 필수 및 권고, 재량 등의 설명, 메타데이터 요소 값의 형식 등의 정보를 갖는다.

메타데이터 스키마는 그 목적에 따라 다양한 표준이 존재한다.

- 정보자원의 검색, 관리, 서비스하기 위한 스키마
 - 정보자원을 생산하는 주제 분야별로 다양한 스키마가 존재한다. 지구 과학 분야에서 사용하는 DIF와 사회과학분야에서 사용하는 DDI, 도서관 분야에서 사용하는 MARC-XML 등이 대표적인 스키마이다.
- 메타데이터(정보자원)을 교환하기 위한 스키마
 - METS는 표준화된 형식으로 메타데이터를 교환하기 위한 대표적인 스키마이다.
- 정보자원을 출판하기 위한 스키마
 - DataCite나 CrossRef에서 연구 데이터(research data)와 논문의 출판을 위한 메타데이터 스키마를 제안하고 있다.
- 정보자원을 관리하는 리포지터리(repository) 스키마 등
 - DCMI에서는 웹 환경에서 접근 가능한 모든 것을 자원(resource)으로 정의하고 있다. 따라서 정보자원을 수집, 관리, 배포하는 기관 리포지터리(institutional repository, IR) 또한 자원이다. 따라서 IR에 대한 설명(IR 이름, IR 운영기관 등)도 메타데이터로 기술될 수 있다. IR을 설명하기 위한 메타데이터 스키마(메타데이터 요소를 어떻게 구성할지 정의한 것) 또한 다양하다. 호주의 RIF-CS나 re3data.org가 IR을 설명하기 위한 대표적인 메타데이터 스키마이다.

1.6 메타데이터 표준의 필요성

메타데이터 표준의 필요성을 설명하기 위해, 정확한 비유는 아니지만 우리가 사용하는 ‘언어’를 이용해 설명한다. 영어가 만국 공통어로 사용되기에 우리는 영어를 사용한다. 이는 영단어의 의미를 서로 이해하고 있기 때문에 가능하다.

언어로 의사소통을 하는 것과 같이 정보자원(데이터)을 설명한 메타데이터를 제 3자가 동일한 의미로 이해하기 위해서는 정보자원을 설명하는 용어(메타데이터)도 표준화가 되어 있어야 한다. 공통된 용어를 사용할 때 우리는 약속된 의미로 데이터를 이해할 수 있다. 다시 말해, 메타데이터 표준이 필요한 것은 정보자원에 대한 데이터(메타데이터) 작성 시 약속된 의미의 용어(메타데이터 요소)를 사용하기 위해서이다.

이러한 메타데이터 표준과 관련된 다양한 활동이 있지만 그 중 으뜸은 [DCMI\(Dublin Core® Metadata Initiative\)](#) 활동이다. DCMI에서는 웹 환경에서 접근 가능한 모든 것을 자원(resource)로 정의하고 있다. 이러한 자원을 기술(describe)하기 위한 메타데이터 요소 15개를 “[Dublin Core Metadata Element Set\(DC\)](#)”으로 정의하여 사용을 권고하고 있다. 이 요소들은 [ISO Standard 15836:2009](#), [ANSI/NISO Standard Z39.85-2012](#) 표준으로 등록되어 있다.

DC에 정의되어 있는 ‘title’ 요소의 경우, title이라는 단어만 놓고 보면 직함, 출판물, 제목, 칭호, (스포츠)타이틀 등 다양한 의미를 갖게 된다. 하지만 <표 1>과 같이 DC에서는 ‘자원에 주어진 이름’으로 정의되어 있다. 부가적으로 ‘자원의 이미 알려진 이름’으로 설명되어 있다.

Term Name: title	
URI:	http://purl.org/dc/elements/1.1/title
Label:	Title
Definition:	A name given to the resource.
Comment:	Typically, a Title will be a name by which the resource is formally known.

표 4 DCMI에서 정의한 메타데이터 요소 title의 의미 (출처: <http://bit.ly/1EWtC94>)

우리가 DC에서 정의되어 있는 ‘title’요소를 사용하여 메타데이터를 작성할 경우, DC에서 정의한 ‘title’요소의 의미를 사용하는 것이다. 따라서 제 3자가 메타데이터 ‘title’이라는 요소를 보고 ‘자원의 이름’으로 이해할 수 있게 된다. 이쯤 되면 궁금증이 생길 것이다.

‘DC말고도 메타데이터 표준이 많이 있을 텐데? title 요소가 DC에서 정의한 title이란 것을 어떻게 표현할 수 있을까?’

당연한 질문이다. 그래서 필요한 것이 네임스페이스(namespace, 이름공간)이다. 네임스페이스에 대해서는 아래에서 설명한다.

1.7 메타데이터 요소의 사용과 이름공간(namespace)

실세계에서 메타데이터 요소가 사용되는 방법은 아주 다양하다. 정보자원의 설명을 관계형 데이터베이스(relational database, RDB)에 저장하는 경우에도 사용되며, XML 파일 형태로 메타데이터를 보관하거나 타 기관과 메타데이터를 교환할 때도 사용된다.

● 데이터베이스의 테이블 스키마

정보자원에 대한 정보(데이터에 대한 데이터)를 RDB에 저장할 경우, 테이블을 설계해야 한다. 설계되는 테이블의 컬럼(요소) 하나, 하나를 통틀어 ‘테이블 스키마’라 부른다. 참고로 RDB에서 말하는 ‘스키마’는 [‘XML 스키마’](#)와는 다르다. 정보자원을 메타데이터를 RDB에 담기 위해 메타데이터 요소 이름을 ‘컬럼 이름’으로 사용할 수 있다.

● XML 요소

XML 문서에서 메타데이터 요소가 사용될 경우에는 메타데이터 요소의 의미를 정확하게 표현하기 위해 ‘이름공간’을 사용한다. 예를 들어, DCMI에서 정의한 메타데이터 코어 요소를 XML 문서에 사용할 경우, 문서의 상단에 DCMI에서 정의한 메타데이터 요소를 지칭하는 이름공간(네임스페이스, 접두어)을 선언하고, XML 엘리먼트(element, 요소)에 ‘’와 ‘콜론(:) 혹은 마침표(.)’를 붙여 사용한다.

다음과 같이 네임스페이스가 사용되는 경우를 자주 보게 된다. <표2>는 ‘네임스페이스(dc)’와 ‘콜론(:)’을 사용하여 메타데이터를 기술한 xml 사례를 보여준다.

```

<?xml version="1.0"?>
<!-- dc 메타데이터 요소를 사용하기 위해 dc 네임스페이스 선언 -->
<dc:descriptionSet
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcx="http://purl.org/dc/xml/"
  xmlns:my="http://example.org/myapplication/">

  <my:webPageDescription>
    <dc:title> <!-- dc 네임스페이스 사용 -->
      <dcx:valueString>DCMI Home Page</dcx:valueString>
    </dc:title>
  </my:webPageDescription>

</dc:descriptionSet>

```

표 5 네임스페이스를 이용한 메타데이터 요소 사용 사례
출처: <http://dublincore.org/documents/dc-xml/>

네임스페이스를 사용하여 메타데이터 요소를 사용하는 것이 생소하게 보일 것이다. 네임스페이스는 이미 정의되어 사용되는 메타데이터 요소를 사용할 때 필요하다. 특히, 여러 분야에서 정의된 요소명(메타데이터 엘리먼트 이름)이 같을 경우 요소의 충돌이 발생하고, 정확한 의미 전달이 어렵게 된다. 예를 들어, 건축분야에서 사용되는 ‘강도’라는 용어(메타데이터 요소)와 경찰행정 분야에서 사용되는 ‘강도’라는 용어는 그 형태는 같으나 의미하는 것이 다르다. 전자는 시멘트 등의 ‘단단하기 정도’를 의미할 수 있으며, 후자는 ‘범죄자’를 의미할 수 있다. 따라서 메타데이터를 작성할 때 건축분야의 메타데이터 요소를 사용하고자 할 경우, 해당 분야의 네임스페이스를 선언하고 ‘강도’를 사용하게 되면 그 의미가 명확해 진다.

XML 문서에서는 엘리먼트 이름으로 한글을 허용하지 않는다. 따라서 <표3>에서 제시한 문서는 유효한 XML문서가 아니나 이해를 돕기 위해 사용하였다.

```
<?xml version="1.0"?>
<!-- 건축분야 메타데이터 요소를 사용하기 위해 네임스페이스 선언 -->
<dcx:descriptionSet
  xmlns:건축="http://construction.com/terms/"
  xmlns:dcx="http://purl.org/dc/xml/"
  xmlns:my="http://example.org/myapplication/">

  <my:webPageDescription>
    <건축:강도> <!-- 건축 네임스페이스 사용 -->
      <dcx:valueString>DCMI Home Page</dcx:valueString>
    </건축:강도>
  </my:webPageDescription>

</dcx:descriptionSet>
```

표 6 네임스페이스를 이용한 메타데이터 요소 사용 사례
출처: <http://dublincore.org/documents/dc-xml/> 사례를 수정하여 사용

XML과 네임스페이스에 대한 보다 자세한 설명은 [w3schools의 웹문서](#)를 참고하기 바란다.

앞서 설명한 메타데이터의 사용 사례 외에도 다양한 분야에서 메타데이터를 사용한다. 예를 들어, 메타데이터 교환을 위해 [JSON \(JavaScript Object Notation\)](#) 형식의 문서 생성 또는 [Turtle](#)이나 [N3\(Notation 3\)](#) 언어로 [RDF 트리플\(온톨로지\)](#)을 생성할 때도 사용될 수 있다. <표4>, <표5>, <표6>, <표7>은 각각의 활용사례를 보여준다.

상당수의 기관에서 Open API 서비스를 제공할 경우, JSON 형식의 값을 응답 형식으로 사용하고 있다. 이렇듯 잘 설계된 메타데이터 요소는 사람과 시스템 간의 이해를 돕는다. 즉, 표준화된 메타데이터 요소가 사람만의 이해를 위해 필요한 것은 아니라는 것이다. 컴퓨터 시스템의 이해도 돕는다. 이를 통해 시스템 간에 상호운영성을 높일 수 있다.

```
{"book": {"dc:title": "가을이 흐른다","dc:creator": "김선태" }}
```

표 7 JSON에서 메타데이터 요소 사용 사례


```

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ex: <http://example.org/stuff/1.0/> .

<http://www.w3.org/TR/rdf-syntax-grammar>
  dc:title "RDF/XML Syntax Specification (Revised)" ;
  ex:editor [
    ex:fullname "Dave Beckett";
    ex:homePage <http://purl.org/net/dajobe/>
  ] .

```

표 8 Turtle에서 메타데이터 요소 사용 사례
(출처: <http://www.w3.org/TeamSubmission/turtle/>)

```

@prefix dc: <http://purl.org/dc/elements/1.1/>.

<http://en.wikipedia.org/wiki/Tony_Benn>
  dc:title "Tony Benn";
  dc:publisher "Wikipedia".

```

표 9 N3에서 메타데이터 요소 사용 사례
(출처: <https://en.wikipedia.org/wiki/Notation3>)

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://en.wikipedia.org/wiki/Tony_Benn">
    <dc:title>Tony Benn</dc:title>
    <dc:publisher>Wikipedia</dc:publisher>
  </rdf:Description>
</rdf:RDF>

```

표 10 RDF에서 메타데이터 요소 사용예
(출처: <https://en.wikipedia.org/wiki/Notation3>)

2. 과학데이터 유통을 위한 메타데이터

2.1 목적과 범위

이 문서는 연구자나, 연구자를 지원하는 업무 담당자, 도서관이나 데이터센터 직원, 리포지터리(repository) 운영자 등에게 과학데이터를 유통하기 위한 유통구조와 표준 메타데이터 스키마를 설명하기 위해 작성되었다. 국가 과학데이터 센터에서는 과학데이터를 모든 연구자들이 쉽게 공유하여 활용할 수 있도록 컬렉션, 아이템, 파일의 유통구조로 관리하고자 하며, 이들에 대해 각각의 메타데이터를 정의하여 본 문서에서 설명하고자 한다.

2.2 과학데이터 유통 구조

다양한 과학분야 기관들로부터 여러 종류의 과학데이터를 효과적으로 공유하고 관리하고자 국가 과학데이터 센터에서는 과학데이터를 컬렉션-아이템-파일의 구조를 구성하여 유통하고자 한다.

컬렉션의 관리자는 컬렉션을 생성하여 원하는 목적별(연구사업관리, 과제관리, 주제분야관리 등)로 과학데이터를 관리할 수 있다. 컬렉션을 생성하는 경우, 컬렉션의 하위 객체인 아이템을 생성할 수 있는 사용자나 사용자그룹을 지정하여야만 한다.

컬렉션 하위객체로서 생성되는 아이템은 공유와 활용을 위한 과학데이터 파일의 묶음이라고 할 수 있다. 따라서 인용을 위한 DOI(Digital Object Identifier)가 부여될 수 있다. 단, DOI를 발급하면 그 시점부터 아이템과 그 하위 객체인 파일들은 수정이 불가능하다. 이는 인용 대상이 변경되는 것을 방지하기 위함이다.

파일은 공유 활용의 가치가 있는 개별 단위의 과학데이터로서 최종적으로 연구자들이 활용하는 대상이 된다. 파일 그 자체로도 공유 활용이 가능하며, DOI 부여도 가능하다

	개념도	정의	메타데이터 성격
신스레코		연구사업, 조직, 과제별로 연구데이터를 관리하기 위한 관리 목적의 아이템 그룹	<ul style="list-style-type: none"> • 관리 목적 • 주제 분야: 대부분 • 관련 객체 표시 • 관련 사용자 정의
아이템		공유 활용성을 높이기 위해 활용목적별로 데이터를 그룹화한 파일의 묶음	<ul style="list-style-type: none"> • 관리 / 플랫폼 내 공유 목적 • 주제 분야: 중분류 • 관련 객체 표시 • 관련 사용자 정의 • 관련 접근권한 정의
파일		공유 활용의 가치가 있는 개별 단위의 과학데이터	<ul style="list-style-type: none"> • 공유/활용/보존 목적 • 주제 분야: 소분류 • DDC 분류 필요(출판) • 보존/관리 메타-정의 • 관련 저장권 정의 • 관련 접근권한 정의

그림 2 과학데이터를 유통하기 위한 컬렉션-아이템-파일 구조에 대한 설명

2.3 과학데이터 구조 별 정의사례

본 장에서는 과학데이터의 유통구조를 쉽게 이해할 수 있도록 생성 가능한 구조 별로 예시를 작성하여 설명하고자 한다.

아래 그림에서 컬렉션 C1은 '과학데이터 거버넌스 체계 수립' 사업을 관리하기 위해 생성되었으며, 사용자 홍길동, 이순신, 유관순에게 아이템 생성 권한을 주었다. 특히, 유관순에게는 아이템의 삭제 권한까지 주어 불필요한 아이템을 삭제하여 관리하도록 하였다.

아이템 I1-1은 '과학데이터 거버넌스 체계 수립' 사업 내에 '메타데이터 표준화' 업무 영역의 자료를 모아놓기 위해 생성되었다. 아이템에 대한 접근 기간은 2015년 1월 11일부터 1년 간이며, 일반 사용자들에게 1만원의 데이터 제공요금을 받는 조건으로 공개하고 있다.

	개념도	필수 메타데이터 예시												
컬렉션		<p><'컬렉션 C1'의 메타데이터 예시></p> <table border="1"> <tr><td>Collection ID</td><td>Co_XXXXXX1</td></tr> <tr><td>Collection Name</td><td>과학데이터 거버넌스 체계 수립</td></tr> <tr><td>Collection Dates</td><td>2015년 01월 01일 (생성일)</td></tr> <tr><td>Access User</td><td>홍길동 (과학기술인등록번호: XXXXXX - 아이템 생성), 이순신 (과학기술인등록번호: XXXXXX - 아이템 생성), 유관순 (과학기술인등록번호: XXXXXX - 아이템 생성 및 삭제)</td></tr> <tr><td>Related Object*</td><td>IT_XXXXXX1 (하위객체), IT_XXXXXX2(하위객체)</td></tr> <tr><td>Collection Subject*</td><td>ee-정보/통신 (국가과학기술표준분류체계 - 대분류)</td></tr> </table>	Collection ID	Co_XXXXXX1	Collection Name	과학데이터 거버넌스 체계 수립	Collection Dates	2015년 01월 01일 (생성일)	Access User	홍길동 (과학기술인등록번호: XXXXXX - 아이템 생성), 이순신 (과학기술인등록번호: XXXXXX - 아이템 생성), 유관순 (과학기술인등록번호: XXXXXX - 아이템 생성 및 삭제)	Related Object*	IT_XXXXXX1 (하위객체), IT_XXXXXX2(하위객체)	Collection Subject*	ee-정보/통신 (국가과학기술표준분류체계 - 대분류)
Collection ID	Co_XXXXXX1													
Collection Name	과학데이터 거버넌스 체계 수립													
Collection Dates	2015년 01월 01일 (생성일)													
Access User	홍길동 (과학기술인등록번호: XXXXXX - 아이템 생성), 이순신 (과학기술인등록번호: XXXXXX - 아이템 생성), 유관순 (과학기술인등록번호: XXXXXX - 아이템 생성 및 삭제)													
Related Object*	IT_XXXXXX1 (하위객체), IT_XXXXXX2(하위객체)													
Collection Subject*	ee-정보/통신 (국가과학기술표준분류체계 - 대분류)													
아이템		<p><'아이템 I1-1'의 메타데이터 예시></p> <table border="1"> <tr><td>Item ID</td><td>IT_XXXXXX1</td></tr> <tr><td>Item Name</td><td>메타데이터 표준화</td></tr> <tr><td>Item Dates</td><td>2015년 01월 11일 (생성일)</td></tr> <tr><td>Item Access Policy</td><td>조건부 (2015년 01월 11일 ~ 2016년 1월 10일, 1만원, 민간사용자 요금지불)</td></tr> <tr><td>Related Object</td><td>Co_XXXXXX1 (상위객체), F_XXXXXX1(하위객체), F_XXXXXX2(하위객체)</td></tr> <tr><td>Item Subject</td><td>ee01. 정보이론 (국가과학기술표준분류체계 - 중분류)</td></tr> </table>	Item ID	IT_XXXXXX1	Item Name	메타데이터 표준화	Item Dates	2015년 01월 11일 (생성일)	Item Access Policy	조건부 (2015년 01월 11일 ~ 2016년 1월 10일, 1만원, 민간사용자 요금지불)	Related Object	Co_XXXXXX1 (상위객체), F_XXXXXX1(하위객체), F_XXXXXX2(하위객체)	Item Subject	ee01. 정보이론 (국가과학기술표준분류체계 - 중분류)
Item ID	IT_XXXXXX1													
Item Name	메타데이터 표준화													
Item Dates	2015년 01월 11일 (생성일)													
Item Access Policy	조건부 (2015년 01월 11일 ~ 2016년 1월 10일, 1만원, 민간사용자 요금지불)													
Related Object	Co_XXXXXX1 (상위객체), F_XXXXXX1(하위객체), F_XXXXXX2(하위객체)													
Item Subject	ee01. 정보이론 (국가과학기술표준분류체계 - 중분류)													
파일														

* 표시는 필수가 아닌 예시 요소임

그림 3 컬렉션 - 아이템 간 메타데이터 정의에 대한 예시

컬렉션은 하위 객체로 아이템뿐 아니라 컬렉션을 가질 수 있다. 컬렉션 C2는 '국가 과학데이터 공유활용'을 위한 기술 자료를 관리하기 위해 생성되었다. 그리고 그 하위 컬렉션 C3는 그 하위 객체로서 공유활용 기술자료 중 '국가 과학데이터 공유활용 플랫폼'에 활용할 자료들만 별도로 관리하기 위해서 생성되었다.

하위 객체인 C3는 상위 객체의 접근 사용자 내에서 접근권한을 재부여할 수 있다. 컬렉션의 범위가 너무 커서 관리가 어려울 경우에는 하위 객체로 컬렉션을 생성하여 사용자의 범위를 줄여서 관리할 수 있다.

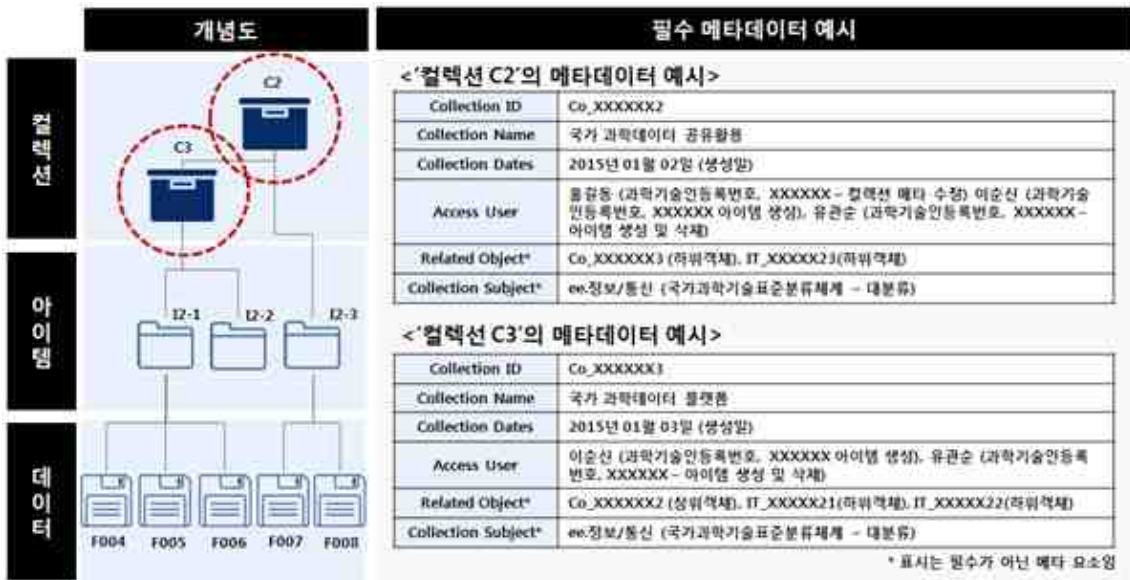


그림 4 컬렉션 - 컬렉션 간 메타데이터 정의에 대한 예시

아이템은 하위객체로 파일을 갖지 않고, 기술된 정보만으로 존재할 수 있다. 아이템 I2-2의 경우, '메타데이터 가이드라인'이라는 문서에 대한 관리를 위해 위치정보 및 접근조건에 대해 기술하고 있다.

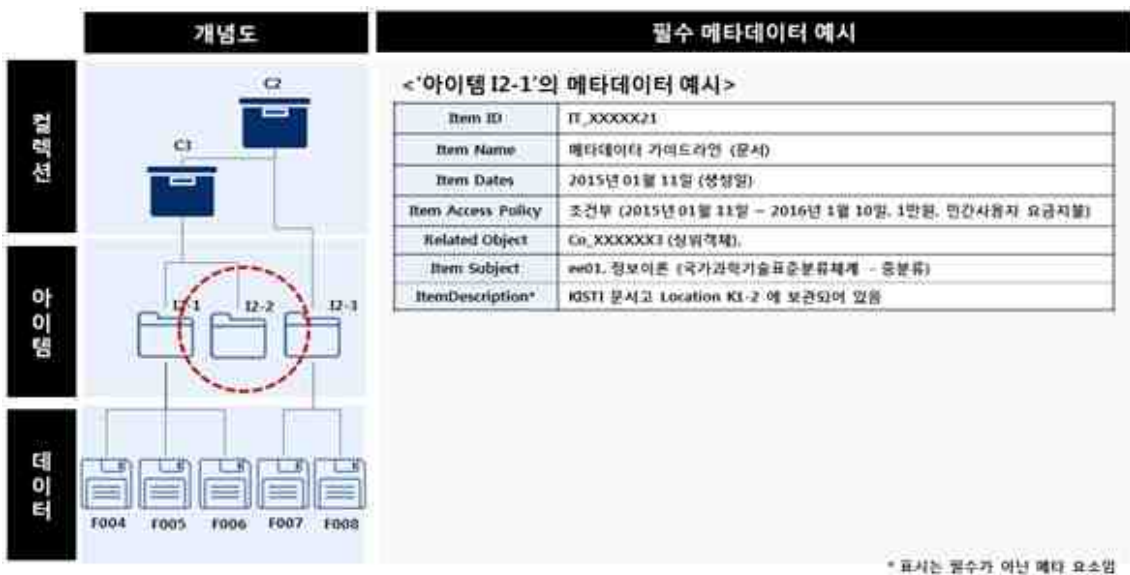


그림 5 하위객체가 없는 아이템의 메타데이터 정의에 대한 예시

2.4 컬렉션 메타데이터

2.4.1 스키마 정의

- 필수 영역

컬렉션을 등록, 생성하기 위해 필수적으로 요구되는 메타데이터는 다음과 같다.

메타데이터 요소	정의	발생	생성주체	참조
CollectionID	컬렉션을 식별하기 위한 ID	M, 1	자동	
CollectionName	컬렉션의 명칭	M, 1	등록자	
CollectionDates	해당 컬렉션과 연관된 날짜정보	M, 1-n	등록자	
- datesType	(생성일,제출일,발행일,승인일,유효기간,기타)으로 구분하여 입력	M, 1	등록자	
CollectionDescription	해당 컬렉션에 대한 부연 설명	M, 1	등록자	
- descriptionType	(개요, 목적, 해당분야 중요성, 기타)로 구분하여 입력	M, 1	등록자	
AccessUser	해당 컬렉션에 대한 접근 가능 사용자 이름	M, 1-n	등록자	
- userIdentifier	해당 컬렉션에 대한 접근 가능 사용자 식별자 (과학기술인 등록번호)	M, 1	등록자	
- accessType	접근 권한 유형은 (컬렉션 메타 수정, 아이템 생성, 아이템 생성/삭제)으로 구분하여 입력	M, 1	등록자	

표 11 컬렉션 메타데이터 필수영역 정의

- 선택 영역

컬렉션을 관리하기 위해 추가적으로 요구되는 메타데이터는 다음과 같다.

메타데이터 요소	정의	발생	생성주체	참조
CollectionSubject	해당 컬렉션에 대한 주제분야	0, 0-n	등록자	
- subjectScheme	(과학기술표준분류 대분류 기준)으로 구분하여 입력	M, 1	등록자	
CollectionOrganization	해당 컬렉션과 관련된 기관	0, 0-n	등록자	
- organizationType	해당 컬렉션에 대한 (담당기관, 출처기관, 사용기관, 기타) 로 구분	M, 1	등록자	
RelatedObject	해당 컬렉션과 연관된 객체의ID (컬렉션 ID or 아이템 ID)	0, 0-n	등록자	
- relationType	(하위객체, 상위객체)를 구분하여 입력	M, 1	등록자	

표 12 컬렉션 메타데이터 선택영역 정의

2.4.2 요소 정의 및 작성 방법

컬렉션 메타데이터의 각 요소들은 다음과 같이 사용된다.

요 소 명	CollectionID
관련 요소	[None]
상세 설명	컬렉션을 식별하기 위해 부여되는 고유한 ID
작성 방법	시스템에서 자동적으로 발급되어 부여함
작성 예시	Co000001

요 소 명	CollectionName
관련 요소	[None]
상세 설명	컬렉션의 이름
작성 방법	컬렉션을 생성한 사용자가 자유롭게 기술함 연구사업 또는 과제별로 관리를 위해 컬렉션을 생성할 수 있으며, 이름을 통해 컬렉션의 생성 목적을 알 수 있도록 작성하는 것을 권장함
작성 예시	과학데이터 거버넌스 체계 수립 사업

요 소 명	CollectionDates
관련 요소	[하위요소] datesType
상세 설명	해당 컬렉션과 관련된 생성, 유효기간 등의 일자 정보
작성 방법	컬렉션을 생성한 일자를 자동적으로 입력함(필수) 컬렉션에 대한 접근, 사용기간을 유효기간으로 입력할 수 있음(선택) 추가적으로 입력하고 싶은 일자 정보를 기타 유형을 선택하여 입력할 수 있음(선택)
작성 예시	2015년 10월 18일 (생성일) 2015년 10월 18일 ~ 2016년 10월 17일 (유효기간) 2015년 08월 01일 ~ 2015년 10월 31일 (기타, 해당 사업기간)

요 소 명	datesType
관련 요소	[상위요소] CollectionDates
상세 설명	해당 컬렉션과 관련된 일자 정보의 유형
작성 방법	(생성일, 유효기간, 기타) 중 선택하여 입력 '기타'를 선택한 경우, 유형의 값을 추가적으로 입력
작성 예시	2015년 10월 18일 (생성일) 2015년 10월 18일 ~ 2016년 10월 17일 (유효기간) 2015년 08월 01일 ~ 2015년 10월 31일 (기타, 해당 사업기간)

요 소 명	CollectionDescription
관련 요소	[하위요소] descriptionType
상세 설명	해당 컬렉션에 대한 설명을 자유롭게 기술
작성 방법	개요, 목적, 해당분야 내 컬렉션의 가치 등을 자유롭게 기술하여 작성함
작성 예시	과학데이터를 공유활용할 수 있도록 거버넌스 체계를 구축함으로써 다방면의 연구활동에 이바지함 (해당분야 중요성)

요 소 명	descriptionType
관련 요소	[상위요소] CollectionDescription
상세 설명	해당 컬렉션 설명에 대한 유형
작성 방법	(개요, 목적, 해당분야 중요성, 기타)로 구분하여 입력 '기타'를 선택한 경우, 유형의 값을 추가적으로 입력
작성 예시	(정해진 값 중에 선택)

요 소 명	AccessUser
관련 요소	[하위요소] userIdentifier, accessType
상세 설명	해당 컬렉션에 대한 접근이 가능하도록 승인된 사용자명 또는 그룹명
작성 방법	컬렉션 접근이 허가된 사용자에 대한 정보와 접근 권한에 대한 정보를 기입
작성 예시	홍길동

요 소 명	userIdentifier
관련 요소	[상위요소] AccessUser
상세 설명	해당 컬렉션에 대한 접근 승인된 사용자의 식별자
작성 방법	사용자에 대한 식별자는 과학기술인등록번호를 사용 사용자 그룹에 대한 식별자는 플랫폼에서 관리하는 식별코드를 사용
작성 예시	홍길동 (123457)

요 소 명	accessType
관련 요소	[상위요소] AccessUser
상세 설명	해당 컬렉션에 대한 접근 권한에 대한 유형 분류
작성 방법	접근 권한 유형에는 (컬렉션 메타 수정, 아이템 생성, 아이템 생성/삭제)이 있음
작성 예시	홍길동 (1324567, 아이템 생성 권한)

요 소 명	CollectionSubject
관련 요소	[하위요소] subjectScheme
상세 설명	해당 컬렉션에 대한 주제분야
작성 방법	과학기술표준분류체계의 대분류로 구분하는 것을 권장함 그외의 분류체계로도 주제분야를 입력할 수 있음
작성 예시	ee.정보/통신

요 소 명	subjectScheme
관련 요소	[상위요소] CollectionSubject
상세 설명	해당 컬렉션에 대한 주제분야의 스키마
작성 방법	과학기술표준분류체계 또는 그 외의 분류체계
작성 예시	ee.정보/통신 (과학기술표준분류체계)

요 소 명	CollectionOrganization
관련 요소	[하위요소] organizationType
상세 설명	해당 컬렉션과 관련된 기관
작성 방법	해당 컬렉션과 관련된 기관의 명칭을 기입함 NTIS에 등록된 기관정보를 연계하여 검색하는 기능을 구현함 등록되지 않은 기관정보는 별도로 기재하여 추가적으로 등록하도록 함
작성 예시	한국과학기술정보연구원

요 소 명	organizationType
관련 요소	[상위요소] CollectionOrganization
상세 설명	해당 컬렉션에 대한 (담당기관, 출처기관, 사용기관, 기타) 로 구분
작성 방법	해당 컬렉션에 대해 어떤 책임이나 역할을 갖는 기관인지 구분하여 입력 해당 컬렉션의 관리 및 작성을 담당하는 담당기관인지, 아니면 컬렉션 데이터의 출처기관인지, 아니면 컬렉션을 주로 이용하는 사용기관인지 구분하여 입력 그 외의 경우는 기타로 입력하여 텍스트값으로 기입
작성 예시	한국과학기술정보연구원 (담당기관)

요 소 명	RelatedObject
관련 요소	[하위요소] relationType
상세 설명	해당 컬렉션과 연관된 객체의 ID (컬렉션 ID 또는 아이템 ID)
작성 방법	컬렉션과 연관되어 있는 컬렉션 ID 또는 아이템 ID를 기입
작성 예시	IT_XXXXXX1, IT_XXXXXX2

요 소 명	relationType
관련 요소	[상위요소] RelatedObject
상세 설명	해당 컬렉션과 기술한 객체와의 관계에 대한 설명
작성 방법	컬렉션이 (컬렉션/아이템)을 하위로 갖으면, (컬렉션ID/아이템 ID)를 입력하고 relationType에서 '하위객체' 선택 컬렉션이 상위 컬렉션 밑에 존재할 경우, (컬렉션ID)를 입력하고 relationType에서 '상위객체' 선택
작성 예시	IT_XXXXXX1 (하위객체), IT_XXXXXX2(하위객체)

2.4.3 XML 작성 예시

다음은 XML형식으로 작성된 컬렉션 메타데이터의 작성 예시이다.

```

<CollectionID> Co.1234567 </CollectionID>
<CollectionName> 과학데이터 거버넌스 체계 수립 사업 </CollectionName>
<CollectionDates datesType = "생성일"> 20151018 </CollectionDates>
<CollectionDates datesType = "유효기간"> 20151018to20161017 </CollectionDates>
<CollectionDescription descriptionType = "해당분야 중요성">
"과학데이터를 공유활용할 수 있도록 거버넌스 체계를 구축함으로써 다방면의 연구활동에
이바지함"
</CollectionDescription>
<AccessUser userIdentifier = "123457" accessType = "아이템 생성 권한"> 홍길동
</AccessUser>
<AccessUser userIdentifier = "123458" accessType = "아이템 생성/삭제 권한"> 유관순
</AccessUser>
<CollectionSubject subjectScheme = "과학기술표준분류체계"> ee.정보/통신
</CollectionSubject>
<CollectionOrganization organizationType = "담당기관"> 한국과학기술정보연구원
</CollectionOrganization>
<RelatedObject relationType = "하위객체"> IT_XXXXXX1 </RelatedObject>
<RelatedObject relationType = "하위객체"> IT_XXXXXX2 </RelatedObject>

```

표 13 컬렉션 메타데이터를 XML 형태로 작성한 예시

2.5 아이템 메타데이터

2.5.1 스키마 정의

- 필수 영역

아이템을 등록, 생성하기 위해 필수적으로 요구되는 메타데이터는 다음과 같다.

메타데이터 요소	정의	발생	생성주체	참조
ItemID	아이템을 식별하기 위한 ID	M, 1	자동	
ItemTitle	아이템의 국문 제목	M, 1	등록자	DC
- englishTitle	아이템의 영문 제목	M, 1	등록자	
- additionalTitle	아이템의 부제목 또는 기타	0, 0-n	등록자	
-- titleType	(부제목, 기타) 으로 제목 유형 구분	M, 1	등록자	DC
ItemDates	해당 아이템과 연관된 날짜정보	M, 1-n	등록자	
- datesType	(생성일,제출일,발행일,승인일,유효기간,기타)으로 구분하여 입력	M, 1	등록자	
ItemSubject	해당 아이템에 대한 주제분야	M, 1-n	등록자	
- subjectScheme	(과학기술표준분류, 6T, 국가전략기술, DDC, 기타)로 구분하여 입력	M, 1	등록자	
ItemAccesspolicy	해당 아이템에 대한 접근 정책	M, 1-n	등록자	
- accessDate	해당 아이템에 대한 접근 가능 기간	0, 1	등록자	
- accessCharge	해당 아이템에 대한 요금	0, 1	등록자	
- accessCondition	해당 아이템에 대한 접근 제한 조건	0, 1	등록자	
ItemCreator	아이템을 생산한 책임연구자 국문 이름	M, 1	등록자	DC
- englishName	아이템을 생산한 책임연구자 영문 이름	M, 1	등록자	DC
- nameIdentifier	아이템을 생산한 책임연구자 ID	M, 1-n	등록자	DC
-- nameIdentifierScheme	(과학기술인등록번호,주민등록번호,여권번호)로식별자스키마구분	M, 1	등록자	DC
ItemPublisher	아이템 보유/제공자 (사업 주관기관) 국문 이름	M, 1	등록자	DC
- englishName	아이템 보유/제공자 (사업 주관기관) 영문 이름	M, 1	등록자	
ItemPublicationYear	과학데이터의 출판일자	M, 1	자동입력	DC
ReferenceID	해당 아이템의 DOI	0, 0-n	자동	
DOIIssue	DOI 발급에 대해 (미신청, 신청, 발급완료)로 구분하여 표시	M, 1	자동	

표 14 아이템 메타데이터 필수영역 정의

● 선택 영역

아이템을 관리하기 위해 추가적으로 요구되는 메타데이터는 다음과 같다.

메타데이터 요소	정의	발생	생성주체	참조
RelatedObject	해당아이템과 연관된 객체의ID (컬렉션 ID or 과학데이터 ID)	0, 0-n	등록자	
- relationType	(하위객체, 상위객체)를 구분하여 입력	M, 1	등록자	
ResourceType	해당 아이템의 데이터 유형 분류(하위분류)	0, 0-1	등록자	
- resourceTypeGeneral	해당 아이템의 데이터 유형 분류(상위분류)	0, 0-1	등록자	
RelatedItem	해당 아이템과 연관된 아이템	0, 0-n	등록자	
- relationType	(분석대상아이템, 산출결과아이템, 선후관계아이템, 기타)를 구분하여 입력	M, 1	등록자	
ItemDescription	해당 아이템에 대한 부연 설명	0, 0-n	등록자	
- descriptionType	(개요, 목적, 기타)로 구분하여 입력	M, 1	등록자	

표 15 아이템 메타데이터 선택영역 정의

2.5.2 요소 정의 및 작성방법

아이템 메타데이터의 각 요소들은 다음과 같이 사용된다.

요소명	ItemID
관련 요소	[None]
상세 설명	아이템을 식별하기 위해 부여되는 고유한 ID
작성 방법	시스템에서 자동적으로 발급되어 부여함
작성 예시	IT000001

요 소 명	ItemTitle
관련 요소	[하위요소] englishTitle, additionalTitle
상세 설명	아이템의 제목
작성 방법	아이템을 생성한 사용자가 자유롭게 국문으로 기술함 아이템은 인용이 가능한 객체단위이기 때문에 활용 목적에 따라 아이템을 생성할 수 있으며, 아이템이 생성된 목적을 알 수 있도록 이름을 등록하는 것을 권장함
작성 예시	메타데이터 표준화

요 소 명	englishTitle
관련 요소	[상위요소] ItemTitle
상세 설명	아이템의 영문 제목
작성 방법	아이템의 제목으로 영문으로 기술함
작성 예시	Standardization of Metadata

요 소 명	additionalTitle
관련 요소	[상위요소] ItemTitle [하위요소] titleType
상세 설명	아이템의 추가적인 설명을 위한 부제목 및 다른 언어의 제목
작성 방법	아이템에 대한 이해를 돕기 위해 부제목이 필요한 경우 부제목을 기술함 국문, 영문 외 다른 국가의 언어로 표현이 필요한 경우 타 언어의 제목을 추가적으로 기술함
작성 예시	과학데이터의 공유활용을 위한 메타데이터의 표준화

요 소 명	titleType
관련 요소	[상위요소] additionalTitle
상세 설명	추가적인 제목 기술의 유형
작성 방법	부제목, 기타 유형 중에 선택하여 기입 기타 유형을 선택한 경우, '00 언어 제목' 과 같이 상세한 유형을 기술
작성 예시	부제목

요 소 명	ItemDates
관련 요소	[하위요소] datesType
상세 설명	해당 아이템과 관련된 생성, 유효기간 등의 일자 정보
작성 방법	아이템을 생성한 일자를 자동적으로 입력함(필수) 아이템에 대한 접근, 사용기간을 유효기간으로 입력할 수 있음(선택) 추가적으로 입력하고 싶은 일자 정보를 기타 유형을 선택하여 입력할 수 있음(선택)
작성 예시	2015년 10월 18일 또는 2015년 10월 18일 ~ 2016년 10월 17일

요 소 명	datesType
관련 요소	[상위요소] ItemDates
상세 설명	해당 아이템과 관련된 일자 정보의 유형
작성 방법	(생성일, 유효기간, 기타) 중 선택하여 입력 '기타'를 선택한 경우, 유형의 값을 추가적으로 입력
작성 예시	2015년 10월 18일 (생성일) 2015년 10월 18일 ~ 2016년 10월 17일 (유효기간) 2015년 08월 01일 ~ 2015년 10월 31일 (기타, 해당 사업기간)

요 소 명	ItemSubject
관련 요소	[하위요소] subjectScheme
상세 설명	해당 아이템에 대한 주제분야
작성 방법	과학기술표준분류체계의 중분류로 구분 (권장) 그외의 분류체계로도 주제분야를 입력할 수 있음
작성 예시	ee01. 정보이론

요 소 명	subjectScheme
관련 요소	[상위요소] ItemSubject
상세 설명	해당 아이템에 대한 주제분야의 스키마
작성 방법	과학기술표준분류체계 또는 그 외의 분류체계
작성 예시	ee01. 정보이론 (과학기술표준분류체계)

요 소 명	ItemAccesspolicy
관련 요소	[하위요소] accessDate, accessCharge, accessCondition
상세 설명	해당 아이템에 대한 접근과 관련한 정책 내용을 기술
작성 방법	해당 아이템이 공개인지, 비공개인지, 조건부 제공인지 구별하여 입력 (공개, 제한, 조건부) 값 중에 선택하여 입력
작성 예시	조건부

요 소 명	accessDate
관련 요소	[상위요소] ItemAccesspolicy
상세 설명	해당 아이템이 공개 또는 조건부 공개일 경우, 공개되는 기간을 입력
작성 방법	공개 시작일부터 공개 종료일까지 기간으로 입력
작성 예시	조건부 (2015년 01월 11일 ~ 2016년 1월 10일)

요 소 명	accessCharge
관련 요소	[상위요소] ItemAccesspolicy
상세 설명	해당 아이템이 조건부 공개이고 요금을 요구하는 경우, 금액을 입력
작성 방법	아이템 공개에 대한 요금을 원화로 입력
작성 예시	조건부 (2015년 01월 11일 ~ 2016년 1월 10일, 100,000원)

요 소 명	accessCondition
관련 요소	[상위요소] ItemAccesspolicy
상세 설명	해당 아이템이 조건부 공개일 경우, 상세 제한조건을 자유롭게 기술
작성 방법	일부 대상에게 한정적으로 제한을 하고 있는 경우 또는 일부 목적으로 사용할 경우에만 한정적으로 제한하는 경우, 이를 자유롭게 기술하여 설명
작성 예시	조건부 (2015년 01월 11일 ~ 2016년 1월 10일, 100,000원, 공공기관 외 일반 사용자들에게 제공요금을 청구함)

요 소 명	ItemCreator
관련 요소	[하위요소] englishName, nameIdentifier
상세 설명	아이템을 생산한 책임연구자의 이름
작성 방법	아이템을 생산한 책임연구자의 이름을 국문으로 입력함
작성 예시	홍길동

요 소 명	englishName
관련 요소	[상위요소] ItemCreator
상세 설명	아이템을 생산한 책임연구자의 영문 이름
작성 방법	아이템을 생산한 책임연구자의 이름을 영문으로 입력함 First name과 Last name을 모두 기입하도록 함
작성 예시	홍길동 (Gil-Dong Hong)

요 소 명	nameIdentifier
관련 요소	[상위요소] ItemCreator, [하위요소] nameIdentifierScheme
상세 설명	아이템을 생산한 책임연구자의 고유 식별자
작성 방법	아이템을 생산한 책임연구자의 식별자가 되는 주민등록번호나 과학기술인등록번호 등을 입력함 (과학기술인등록번호를 권장함)
작성 예시	홍길동 (Gil-Dong Hong, K_XXXXX1)

요 소 명	nameIdentifierScheme
관련 요소	[상위요소] nameIdentifier
상세 설명	아이템을 생산한 책임연구자의 고유 식별자 스키마
작성 방법	아이템을 생산한 책임연구자의 식별자가 주민등록번호인지, 과학기술인등록번호인지, 여권번호인지 선택하도록 함
작성 예시	홍길동 (Gil-Dong Hong, K_XXXXX1, 과학기술인등록번호)

요 소 명	ItemPublisher
관련 요소	[하위요소] englishName
상세 설명	아이템을 보유/제공하는 사람 또는 기관의 명칭
작성 방법	해당 데이터를 보유하고 제공하는 권한을 가진 사람 또는 기관의 명칭을 국문으로 기입
작성 예시	한국과학기술정보연구원

요 소 명	englishName
관련 요소	[상위요소] ItemPublisher
상세 설명	아이템 제공자 또는 기관의 영문 명칭
작성 방법	해당 데이터를 보유하고 제공하는 권한을 가진 사람 또는 기관의 명칭을 영문으로 기입
작성 예시	Korea Institute of Science and Technology Information

요 소 명	ItemPublicationYear
관련 요소	[None]
상세 설명	아이템을 출판한 일자
작성 방법	아이템을 공유 활용할 수 있도록 DOI를 부여하여 공개한 일자를 출판일로 기록하여 관리
작성 예시	2015년 10월 21일

요 소 명	ReferenceID
관련 요소	[None]
상세 설명	해당 아이템의 DOI
작성 방법	해당 아이템에 DOI 발급이 요청되면, DOI를 생성하여 부여된 DOI를 기입 (텍스트 형태의 값을 입력)
작성 예시	http://dx.doi.org/XX.XXX1/XXXX.XXXX.XX.X.XXX

요 소 명	DOIIssue
관련 요소	[None]
상세 설명	DOI 발급과 관련한 아이템 현재 상태 (자동입력)
작성 방법	아직 하위객체인 파일들이 등록 및 수정되고 있는 단계이기 때문에 DOI를 미신청한 상태인 경우, '미신청' 상태를 선택하여 기입 (Default값) 하위객체 등록이 완료되어 출판을 위해 DOI 발급을 요청한 경우, '신청' 상태로 변경 DOI 발급이 처리되어 완료되었을 경우, '발급완료'의 상태로 변경
작성 예시	발급완료

요 소 명	RelatedObject
관련 요소	[하위요소] relationType
상세 설명	해당 아이템과 연관된 객체의 ID (컬렉션 ID 또는 파일 ID)
작성 방법	아이템과 연관되어 있는 컬렉션 ID 또는 파일 ID를 기입
작성 예시	Co_XXXXXX1, F_XXXXXX1, F_XXXXXX2

요 소 명	relationType
관련 요소	[상위요소] RelatedObject
상세 설명	해당 아이템과 기술한 객체와의 관계에 대한 설명
작성 방법	아이템이 파일을 하위로 갖으면, 파일 ID를 입력하고 relationType에서 '하위객체' 선택 아이템이 상위 컬렉션 ID를 입력하고 relationType에서 '상위객체' 선택
작성 예시	Co_XXXXXX1 (상위객체), F_XXXXXX1(하위객체), F_XXXXXX2(하위객체)

요 소 명	ResourceType
관련 요소	[하위] resourceTypeGeneral
상세 설명	해당 아이템의 데이터 유형 분류(하위분류)
작성 방법	해당 아이템이 담고 있는 정보의 유형이 어떤 것인지 분류하여 기입 상위 분류가 이미지인 파일은 '사진, 애니메이션, 그림' 등으로 분류 상위 분류가 문서인 파일은 '보고서, 논문, 정기발행물' 등으로 분류 텍스트 타입으로 자유롭게 기술함
작성 예시	보고서 (문서)

요 소 명	resourceTypeGeneral
관련 요소	[상위] ResourceType
상세 설명	해당 아이템의 데이터 유형 분류(상위분류)
작성 방법	해당 아이템이 담고 있는 정보의 유형이 어떤 것인지 상위 레벨에서 분류 유형의 값은 정해진 값 중에서 선택하여 기입 '이미지, 동영상, 소프트웨어, 문서, 음원, 데이터셋, 기타' 중에서 선택
작성 예시	보고서 (문서)

요 소 명	RelatedItem
관련 요소	[하위요소] relationType
상세 설명	해당 아이템과 연관된 아이템
작성 방법	해당 아이템과 관련된 아이템의 식별자를 입력함
작성 예시	ITXXXXXX1, ITXXXXXX2

요 소 명	relationType
관련 요소	[상위요소] RelatedItem
상세 설명	해당 아이템과 기술한 아이템과의 관계에 대한 설명
작성 방법	해당 아이템이 기술하고 있는 아이템의 분석대상이 되는 객체일 경우, '분석대상아이템'을 선택 해당 아이템이 기술하고 있는 아이템으로부터 도출되어 나온 결과 데이터인 경우, '산출결과아이템'을 선택 해당 아이템이 시간적, 공간적 선후관계를 갖는 데이터인 경우, '선후관계아이템'을 선택 기타 다른 관계를 갖고 있는 경우, '기타'를 선택하여 기술
작성 예시	ITXXXXXX1(분석대상아이템, ITXXXXXX2(산출결과아이템)

요 소 명	ItemDescription
관련 요소	[하위요소] descriptionType
상세 설명	해당 아이템에 대한 설명을 자유롭게 기술
작성 방법	개요, 목적 등을 자유롭게 기술하여 작성함
작성 예시	과학데이터를 공유활용할 수 있도록 메타정보를 표준화하여 관리할 수 있도록 체계를 수립 (목적)

요 소 명	descriptionType
관련 요소	[상위요소] ItemDescription
상세 설명	해당 아이템 설명에 대한 유형
작성 방법	(개요, 목적, 기타)로 구분하여 입력 '기타'를 선택한 경우, 유형의 값을 추가적으로 입력
작성 예시	(정해진 값 중에 선택)

2.5.2 XML 작성 예시

다음은 XML형식으로 작성된 아이템 메타데이터의 작성 예시이다.

```

<ItemID> IT.1234567 </ItemID>
<ItemTitle> 메타데이터 표준화
  <englishTitle>Standardization of Metadata </englishTitle>
  <additionalTitle titleType = "부제목"> 과학데이터의 공유활용을 위한 메타데이터의
표준화 </additionalTitle>
</ItemTitle>
<ItemDates datesType = "생성일"> 20151018 </ItemDates>
<ItemDates datesType = "유효기간"> 20151018to20161017 </ItemDates>
<ItemDates datesType = "기타, 해당사업기간"> 20150801to20151031 </ItemDates>
<ItemSubject subjectScheme= "과학기술표준분류체계"> ee01. 정보이론 </ItemSubject>
<ItemAccesspolicy> 조건부
  <accessDate> 20150111to20160110 </accessDate>
  <accessCharge> 100,000 </accessCharge>
  <accessCondition> 공공기관 외 일반 사용자들에게 제공요금을 청구 </accessCondition>
</ItemAccesspolicy>
<ItemCreator> 홍길동
  <englishName> Gil-Dong Hong </englishName>
  <nameIdentifier nameIdentifierScheme = "과학기술인등록번호"> 1234567
</nameIdentifier>
</ItemCreator>
<ItemPublisher> 한국과학기술정보연구원
  <englishName> Korea Institute of Science and Technology Information
  </englishName></ItemPublisher>
<ItemPublicationYear> 20151021 </ItemPublicationYear>
<ReferenceID> http://dx.doi.org/XX.XXX1/XXXX.XXXX.XX.X.XXX </ReferenceID>
<DOIIssue> 발급완료 </DOIIssue>
<RelatedObject relationType="상위객체"> Co_XXXXXX1 </RelatedObject>
<RelatedObject relationType="하위객체"> F_XXXXXX1 </RelatedObject>
<RelatedObject relationType="하위객체"> F_XXXXXX2 </RelatedObject>
<ResourceType resourceTypeGeneral = "문서"> 보고서 </ResourceType>
<ItemDescription descriptionType = "목적"> 과학데이터를 공유활용할 수 있도록
메타정보를 표준화하여 관리할 수 있도록 체계를 수립 </ItemDescription>

```

표 17 아이템 메타데이터를 XML 형태로 작성한 예시

2.6 파일 메타데이터

2.6.1 스키마 정의

- 필수 영역

파일을 등록, 생성하기 위해 필수적으로 요구되는 메타데이터는 다음과 같다.

메타데이터 요소	정의	발생	생성주체	참조
Identifier	과학데이터 식별 ID	M, 1	자동	
Creator	과학데이터를 생산한 책임연구자 국문 이름	M, 1	등록자	
- englishName	과학데이터를 생산한 책임연구자 영문 이름	M, 1	등록자	
- nameIdentifier	과학데이터를 생산한 책임연구자 ID	M, 1-n	등록자	
-- nameIdentifierScheme	(과학기술인등록번호,주민등록번호,여권번호)로식별자스키마구분	M, 1	등록자	
Title	과학데이터의 국문 제목	M, 1	등록자	
- englishTitle	과학데이터의 영문 제목	M, 1	등록자	
- additionalTitle	과학데이터의 부제목 또는 기타	O, 0-n	등록자	
-- titleType	(부제목, 기타) 으로 제목 유형 구분	M, 1	등록자	
Publisher	과학데이터 보유 / 제공자 (사업주관기관) 국문이름	M, 1	등록자	
- englishName	과학데이터 보유 / 제공자 (사업주관기관) 영문이름	M, 1	등록자	
PublicationYear	과학데이터의 출판일자	M, 1	자동입력	
Subject	과학데이터 주제분야	M, 1	등록자	
- subjectScheme	(국제십진분류법-DDC)으로 구분하여 입력	M, 1	등록자	
Date	과학데이터 관련 일자정보	M, 1-n	자동/등록자	
- dateType	(생성일, 제출일, 발행일, 승인일, 기타)으로 구분하여 입력	M, 1	자동/등록자	
AccessPolicy	해당 데이터에 대한 데이터 접근 정책을 (공개, 조건부, 제한)으로 구분하여 선택	M, 1	등록자	
- accessDate	해당 데이터 접근가능기간	O, 0-1	등록자	
- accessCharge	해당 데이터 접근에 대한 요금	O, 0-1	등록자	
- accessParty	해당 데이터 접근 가능 대상자	O, 0-1	등록자	
- accessCondition	해당 데이터 접근 가능 조건	O, 0-1	등록자	

표 18 파일 메타데이터 필수영역 정의

● 권고 영역

효과적인 파일의 운영·관리하기 위해 국가과학데이터센터에서 제출을 권고하는 파일 메타데이터는 다음과 같다.

메타데이터 요소	정의	발생	생성주체	참조
Project	과학데이터 생산과 관련된 활동/과제	M, 1	자동	
Collection	해당 과학데이터의 컬렉션 ID	0, 0-1	등록자	
Item	해당과학데이터의아이템ID	0, 0-1	자동	
DataCoverage	해당 과학데이터 수집기간 또는 지역의 범위	0, 0-1	등록자	
- coverageType	(수집기간, 수집지역) 구분하여 입력	0, 0-n	등록자	
DataQualityLV	해당 과학데이터에 대한 품질 정보	M, 1	등록자	
Contributor	과학데이터 생성 및 제공에 기여한 사람	0, 0-1	자동	
- contributorType	(수집자, 데이터 관리자, 편집자, 주관기관, 프로젝트리더, 프로젝트 멤버, 등록기관, 연구자, 기타) 구분하여 입력	0, 0-n	등록자	
- contributorName	과학데이터 생성 및 제공에 기여자의 이름	M, 1	등록자	
-- nameIdentifier	과학데이터 생성 및 제공에 기여자 ID	M, 1	등록자	
--- nameIdentifierScheme	(과학기술인등록번호, 주민등록번호, 여권번호)로식별자스키마구분	M, 1	등록자	
Language	과학데이터의 주 사용언어	M, 1	등록자	
ResourceType	해당 과학데이터의 유형 분류(하위분류)	0, 0-1	등록자	
- resourceTypeGeneral	해당 과학데이터의 유형 분류(상위분류)	0, 0-1	등록자	
ReferenceID	해당 과학데이터의 등록된 다른 식별자	M, 1	등록자	
- referenceIDType	(DOI, ISAN, ISBN, ISRC, ISSN, ISTE, ISNI) 구분하여 입력	0, 0-n	등록자	
Size	과학데이터의 크기 (Bytes)	M, 1	등록자	
Format	과학데이터의 형태 (자유롭게 기술)	0, 0-1	등록자	
Description	해당 과학데이터에 대한 부연 설명	0, 0-1	등록자	
- descriptionType	(개요, 목차, 목적, 기타)로 구분하여 입력	0, 0-n	등록자	
CreatingApplication	해당 과학데이터를 생산한 어플리케이션 명	M, 1	등록자	
- applicationVersion	해당 어플리케이션의 버전	0, 0-1	등록자	
DetailedDataInfo	해당 과학데이터의 실제 내용에 대한 View	0, 0-1	등록자	
Right	해당 데이터에 대한 권한 관리 정보	0, 0-n	등록자	
- rightType	(CCL, ODbI, 기타)으로데이터라이선스 유형구분	0, 0-1	등록자	
RightStatement	해당 데이터의 지적재산권에 대한 정보	M, 1	등록자	

표 19 파일 메타데이터 권고영역 정의

- 선택 영역

각 과학기술 분야에서 효율적으로 활용하기 위해 필요한 정보들을 기술하기 위해 다음의 파일 메타데이터를 선택적으로 기술한다.

메타데이터 요소	정의	발생	생성주체	참조
Keyword	해당 데이터에 대한 주요 키워드(자유롭게 기술)	0, 0-n	등록자	
AdditionalInfo	해당 데이터에 대한 추가적인 정보	0, 0-n	등록자	

표 20 파일 메타데이터 선택영역 정의

2.6.2 요소 정의 및 작성방법

파일 메타데이터의 각 요소들은 다음과 같이 사용된다.

요소명	Identifier
관련 요소	[None]
상세 설명	파일을 식별하기 위해 부여되는 고유한 ID
작성 방법	시스템에서 자동적으로 발급되어 부여함
작성 예시	F000001

요소명	Creator
관련 요소	[하위요소] englishName, nameIdentifier
상세 설명	파일을 생산한 책임연구자의 이름
작성 방법	파일을 생산한 책임연구자의 이름을 국문으로 입력함
작성 예시	홍길동

요소명	englishName
관련 요소	[상위요소] Creator
상세 설명	파일을 생산한 책임연구자의 영문 이름
작성 방법	파일을 생산한 책임연구자의 이름을 영문으로 입력함 First name과 Last name을 모두 기입하도록 함
작성 예시	홍길동 (Gil-Dong Hong)

요 소 명	nameIdentifier
관련 요소	[상위요소] Creator, [하위요소] nameIdentifierScheme
상세 설명	파일을 생산한 책임연구자의 고유 식별자
작성 방법	파일을 생산한 책임연구자의 식별자가 되는 주민등록번호나 과학기술인등록번호 등을 입력함
작성 예시	홍길동 (Gil-Dong Hong, K_XXXXX1)

요 소 명	nameIdentifierScheme
관련 요소	[상위요소] nameIdentifier
상세 설명	파일을 생산한 책임연구자의 고유 식별자 스키마
작성 방법	파일을 생산한 책임연구자의 식별자가 주민등록번호인지, 과학기술인등록번호인지, 여권번호인지 선택하도록 함
작성 예시	홍길동 (Gil-Dong Hong, K_XXXXX1, 과학기술인등록번호)

요 소 명	Title
관련 요소	[하위요소] englishTitle, additionalTitle
상세 설명	파일의 제목
작성 방법	파일을 생성한 사용자가 자유롭게 국문으로 기술함 파일은 인용이 가능한 객체단위이기 때문에 활용 목적에 따라 아이템을 생성할 수 있으며, 아이템이 생성된 목적을 알 수 있도록 이름을 등록하는 것을 권장함
작성 예시	파일 메타데이터 정의

요 소 명	englishTitle
관련 요소	[상위요소] Title
상세 설명	파일의 영문 제목
작성 방법	파일의 제목으로 영문으로 기술함
작성 예시	Definition of Metadata for Files

요 소 명	additionalTitle
관련 요소	[상위요소] Title [하위요소] titleType
상세 설명	파일의 추가적인 설명을 위한 부제목 및 다른 언어의 제목
작성 방법	파일에 대한 이해를 돕기 위해 부제목이 필요한 경우 부제목을 기술함 국문, 영문 외 다른 국가의 언어로 표현이 필요한 경우 타 언어의 제목을 추가적으로 기술함
작성 예시	파일의 생산, 등록, 접근에 대한 메타정보의 정의

요 소 명	titleType
관련 요소	[상위요소] additionalTitle
상세 설명	추가적인 제목 기술의 유형
작성 방법	부제목, 기타 유형 중에 선택하여 기입 기타 유형을 선택한 경우, '00 언어 제목' 과 같이 상세한 유형을 기술
작성 예시	파일의 생산, 등록, 접근에 대한 메타정보의 정의 (부제목)

요 소 명	Publisher
관련 요소	[하위요소] englishName
상세 설명	파일 제공자 또는 기관의 국문 명칭
작성 방법	해당 데이터를 보유하고 제공하는 권한을 가진 사람 또는 기관의 명칭을 국문으로 기입
작성 예시	한국과학기술정보연구원

요 소 명	englishName
관련 요소	[상위요소] Publisher
상세 설명	파일 제공자 또는 기관의 영문 명칭
작성 방법	해당 데이터를 보유하고 제공하는 권한을 가진 사람 또는 기관의 명칭을 영문으로 기입
작성 예시	Korea Institute of Science and Technology Information

요 소 명	PublicationYear
관련 요소	[None]
상세 설명	파일을 출판한 일자
작성 방법	파일을 공유 활용할 수 있도록 DOI, 또는 그 외의 식별자를 부여하여 공개한 일자를 출판일로 기록하여 관리
작성 예시	2015년 10월 21일

요 소 명	Subject
관련 요소	[하위요소] subjectScheme
상세 설명	해당 파일에 대한 주제분야
작성 방법	과학기술표준분류체계의 소분류로 구분 (권장) 그 외의 분류체계로도 주제분야를 입력할 수 있음
작성 예시	EE9999. 달리 분류되지 않는 정보/통신

요 소 명	subjectScheme
관련 요소	[상위요소] Subject
상세 설명	해당 파일에 대한 주제분야의 스키마
작성 방법	과학기술표준분류체계 또는 그 외의 분류체계
작성 예시	과학기술표준분류체계

요 소 명	Dates
관련 요소	[하위요소] datesType
상세 설명	해당 파일과 관련된 생성, 유효기간 등의 일자 정보
작성 방법	파일을 생성한 일자를 자동적으로 입력함(필수) 파일을 국가 과학데이터 센터에 제출한 일자를 자동적으로 입력함(필수) 파일을 출판한 경우, 출판 발행일을 입력함(선택) 추가적으로 입력하고 싶은 일자 정보를 기타 유형을 선택하여 입력할 수 있음(선택)
작성 예시	2015년 9월 14일

요 소 명	datesType
관련 요소	[상위요소] Dates
상세 설명	해당 파일과 관련된 일자 정보의 유형
작성 방법	(생성일, 제출일, 발행일, 기타) 중 선택하여 입력 '기타'를 선택한 경우, 유형의 값을 추가적으로 입력
작성 예시	2015년 9월 14일 (생성일) 2015년 9월 16일 (제출일) 2015년 10월 1일 (발행일)

요 소 명	Accesspolicy
관련 요소	[하위요소] accessDate, accessCharge, accessParty, accessCondition
상세 설명	해당 파일에 대한 접근과 관련한 정책 내용을 기술
작성 방법	해당 파일이 공개인지, 비공개인지, 조건부 제공인지 구별하여 입력 (공개, 제한, 조건부) 값 중에 선택하여 입력
작성 예시	조건부

요 소 명	accessDate
관련 요소	[상위요소] Accesspolicy
상세 설명	해당 파일이 공개 또는 조건부 공개일 경우, 공개되는 기간을 입력
작성 방법	공개 시작일부터 공개 종료일까지 기간으로 입력
작성 예시	조건부 (2015년 01월 11일 ~ 2016년 1월 10일)

요 소 명	accessCharge
관련 요소	[상위요소] Accesspolicy
상세 설명	해당 파일이 조건부 공개이고 요금을 요구하는 경우, 금액을 입력
작성 방법	파일 공개에 대한 요금을 원화로 입력
작성 예시	조건부 (2015년 01월 11일 ~ 2016년 1월 10일, 무료)

요 소 명	accessParty
관련 요소	[상위요소] Accesspolicy
상세 설명	해당 파일이 조건부 공개이고 공개대상이 제한적인 경우 공개대상을 기입
작성 방법	파일 공개대상이 되는 사람 또는 그룹, 기관 등을 입력 (시스템을 통해 사람 또는 기관을 조회하여 식별코드로 입력함)
작성 예시	조건부 (2015년 01월 11일 ~ 2016년 1월 10일, 무료, KISTI_빅데이터센터)

요 소 명	accessCondition
관련 요소	[상위요소] Accesspolicy
상세 설명	해당 파일이 조건부 공개일 경우, 상세 제한조건을 자유롭게 기술
작성 방법	요금이나 대상 외 다른 조건이 있는 경우, 자유롭게 서술형으로 입력
작성 예시	조건부 (2015년 01월 11일 ~ 2016년 1월 10일, 무료, KISTI_빅데이터센터, 비상업적 목적으로만 활용 가능)

요 소 명	Project
관련 요소	[None]
상세 설명	해당 파일의 생산과 관련된 과제, 사업활동의 명칭
작성 방법	국가 연구사업 또는 연구기관의 연구활동 중에 해당 파일이 생성되었다면, 해당 연구사업 또는 과제, 활동에 대한 명칭을 기입 (연구사업에 대한 정보는 NTIS와 연동하여 가져오고, 이를 검색하여 입력할 수 있도록 기술적으로 지원함)
작성 예시	국가 과학데이터 거버넌스 체계 수립 사업

요 소 명	Collection
관련 요소	[None]
상세 설명	해당 파일의 컬렉션 ID
작성 방법	파일이 등록된 컬렉션의 명칭을 기입 (사용자가 파일을 등록할 수 있는 컬렉션 목록을 검색하여 선택할 수 있도록 기능을 지원함)
작성 예시	국가 과학데이터 거버넌스 체계 수립 사업

요 소 명	Item
관련 요소	[None]
상세 설명	해당 파일의 아이템 ID
작성 방법	파일이 등록된 아이템의 명칭을 기입 (사용자가 파일을 등록할 수 있는 아이템 목록을 검색하여 선택할 수 있도록 기능을 지원함)
작성 예시	메타데이터 표준화

요 소 명	DataCoverage
관련 요소	[하위] coverageType
상세 설명	해당 과학데이터 수집기간 또는 지역의 범위
작성 방법	범위 유형이 수집기간일 경우, 언제부터 언제까지 데이터를 수집하였는지 기간을 기입함. 수집지역일 경우, 지역명(텍스트형식으로 자유롭게 기술)을 기입함
작성 예시	2015년 1월 1일 ~ 2015년 10월 1일 (수집기간) 서울시 (수집지역)

요 소 명	coverageType
관련 요소	[None]
상세 설명	해당 과학데이터의 시간적/공간적 범위 유형
작성 방법	(수집기간, 수집지역) 중 하나를 구분하여 기입
작성 예시	(수집기간 또는 수집지역 중 택일)

요 소 명	DataQualityLV
관련 요소	[None]
상세 설명	해당 과학데이터에 대한 품질 정보
작성 방법	해당 과학데이터에 대해 메타데이터를 상세히 기술한 정보를 기준으로 데이터 품질 수준을 측정함 필수정보를 모두 입력한 경우, LV1 필수정보와 권고사항 일부를 입력한 경우, LV2 필수정보와 권고사항을 모두 입력한 경우, LV3 필수정보와 권고사항, 선택사항을 모두 입력한 경우, LV4
작성 예시	3LV

요 소 명	Contributor
관련 요소	[하위] contributorType, contributorName
상세 설명	과학데이터 생성 및 제공에 기여한 사람
작성 방법	*** Wrapper 요소 ***
작성 예시	

요 소 명	contributorType
관련 요소	[상위] Contributor
상세 설명	해당 파일에 어떠한 방법으로 기여한 사람인지에 대한 유형정보
작성 방법	(수집자, 데이터 관리자, 편집자, 주관기관, 프로젝트리더, 프로젝트 멤버, 등록기관, 연구자, 기타) 구분하여 입력
작성 예시	주관기관

요 소 명	contributorName
관련 요소	[상위] Contributor, [하위] nameIdentifier
상세 설명	과학데이터 생성, 제공 등의 활동에 기여한 사람의 이름
작성 방법	기여유형에 따라 사람의 이름을 기입하고, 하위요소에서 이에 대한 식별자 및 식별자 스키마에 대해 상세 기술함
작성 예시	한국과학기술정보연구원

요 소 명	nameIdentifier
관련 요소	[상위] contributorName, [하위] nameIdentifierScheme
상세 설명	과학데이터 생성 및 제공에 기여자 ID
작성 방법	(과학기술인등록번호, 주민등록번호, 여권번호)로 식별자를 기입
작성 예시	1234567

요 소 명	nameIdentifierScheme
관련 요소	[상위] nameIdentifier
상세 설명	과학데이터 생성 및 제공에 기여자 ID의 스키마
작성 방법	(과학기술인등록번호, 주민등록번호, 여권번호)로 식별자 스키마 구분하여 선택 기입함
작성 예시	1234567 (과학기술인등록번호)

요 소 명	Language
관련 요소	[None]
상세 설명	과학데이터의 주 사용언어
작성 방법	과학데이터에서 주로 사용하고 있는 언어를 기입
작성 예시	한국어

요 소 명	ResourceType
관련 요소	[하위] resourceTypeGeneral
상세 설명	해당 과학데이터의 유형 분류(하위분류)
작성 방법	해당 파일이 담고 있는 정보의 유형이 어떤 것인지 분류하여 기입 상위 분류가 이미지인 파일은 '사진, 애니메이션, 그림' 등으로 분류 상위 분류가 문서인 파일은 '보고서, 논문, 정기발행물' 등으로 분류 텍스트 타입으로 자유롭게 기술함
작성 예시	보고서 (문서)

요 소 명	resourceTypeGeneral
관련 요소	[상위] ResourceType
상세 설명	해당 과학데이터의 유형 분류(상위 분류)
작성 방법	해당 파일이 담고 있는 정보의 유형이 어떤 것인지 상위 레벨에서 분류 유형의 값은 정해진 값 중에서 선택하여 기입 '이미지, 동영상, 소프트웨어, 문서, 음원, 데이터셋, 기타' 중에서 선택
작성 예시	보고서 (문서)

요 소 명	ReferenceID
관련 요소	[하위] referenceIDType
상세 설명	해당 과학데이터의 등록된 다른 식별자
작성 방법	해당 파일이 담고 있는 정보가 DOI, ISSN 등 다른 식별자가 부여된 것이라면 이를 기입함
작성 예시	http://dx.doi.org/XX.XXX1/XXXX.XXXX.XX.X.XXX (DOI)

요 소 명	referenceIDType
관련 요소	[상위] ReferenceID
상세 설명	해당 과학데이터의 등록된 다른 식별자 유형
작성 방법	(DOI, ISAN, ISBN, ISRC, ISSN, ISTC, ISNI) 주어진 값 중에서 선택하여 기입함
작성 예시	http://dx.doi.org/XX.XXX1/XXXX.XXXX.XX.X.XXX (DOI)

요 소 명	Size
관련 요소	[None]
상세 설명	과학데이터의 크기
작성 방법	해당 파일의 크기를 Bytes로 기입함
작성 예시	2,500,000 (Bytes)

요 소 명	Format
관련 요소	[None]
상세 설명	과학데이터의 파일 형태
작성 방법	해당 파일의 형태를 확장자명으로 기입함 (자유롭게 텍스트 값으로 기입)
작성 예시	PPT (Power Point Template)

요 소 명	Description
관련 요소	[하위] descriptionType
상세 설명	해당 과학데이터에 대한 부연 설명
작성 방법	해당 파일에 대한 부연 설명을 개요, 목차, 목적 등 유형에 맞게 자유롭게 텍스트 값으로 기술함
작성 예시	<ol style="list-style-type: none"> 1. 목적 및 범위 2. 메타데이터의 구조 3. 컬렉션 메타데이터 4. 아이템 메타데이터 5. 파일 메타데이터

요 소 명	descriptionType
관련 요소	[상위] Description
상세 설명	해당 과학데이터에 대한 부연 설명 유형
작성 방법	(개요, 목차, 목적, 기타)로 구분하여 입력
작성 예시	목차

요 소 명	CreatingApplication
관련 요소	[하위] applicationVersion
상세 설명	해당 과학데이터를 생산한 어플리케이션 명
작성 방법	해당 파일을 생산한 어플리케이션의 명칭을 텍스트 값으로 기입
작성 예시	Microsoft Office Power Point

요 소 명	applicationVersion
관련 요소	[상위] CreatingApplication
상세 설명	해당 어플리케이션의 버전
작성 방법	해당 파일을 생산한 어플리케이션의 버전 정보를 텍스트 값으로 기입
작성 예시	Standard 2007

요 소 명	DetailedDataInfo
관련 요소	[None]
상세 설명	해당 과학데이터의 실제 내용에 대한 View
작성 방법	해당 파일이 담고 있는 실제 내용을 간략하게 확인할 수 있도록 이미지 형태로 내용을 보여줄 수 있도록 입력

작성 예시	<p>1. 개요</p> <p>전송 프로세스에서 요구하는 메타데이터의 템플릿은 다음과 같음</p>
	<p>IV. 과학데이터 전송을 위한 메타데이터</p> <p>METS 문서에 대한 설명 과학데이터에 대한 설명 국가중립 메타데이터 필수 영역만 적용 전송 목적에 따라 기술, 소스, 좌차, 환경정보에 대한 메타데이터 기입 METS문서를 구성하는 과학데이터의 계층적 구조를 정의</p>

요 소 명	Right
관련 요소	[하위] rightType
상세 설명	해당 데이터에 대한 권한 관리 정보
작성 방법	해당 파일 사용에 대한 권한이 어떻게 되는지 기입함 '저작자 표시, 비영리 사용, 동일조건변경허락 등'의 권한 정보를 기입함
작성 예시	CC-BY-NC-SA

요 소 명	rightType
관련 요소	[상위] Right
상세 설명	해당 데이터에 대한 라이선스 유형 구분
작성 방법	(CCL, ODbI, 기타) 으로 데이터 라이선스 유형 구분
작성 예시	CCL

요 소 명	RightStatement
관련 요소	[None]
상세 설명	해당 데이터의 지적재산권에 대한 정보
작성 방법	지적재산권, 저작권의 소유자, 등록일 등을 텍스트 형태로 기입함
작성 예시	Copyright, John M. Smith, 2011

요 소 명	Keyword
관련 요소	[None]
상세 설명	해당 데이터에 대한 주요 키워드
작성 방법	해당 파일의 내용을 쉽게 이해할 수 있도록 관련된 내용의 키워드를 기입 여러 개의 키워드를 나열하여 입력할 수 있으며, 텍스트 형태로 자유롭게 기술함
작성 예시	메타데이터, 표준화, 메타정보, 과학데이터, 연구데이터

요 소 명	AdditionalInfo
관련 요소	[None]
상세 설명	해당 데이터에 대한 추가적인 정보
작성 방법	분야별 과학적 활용을 위해 필요한 추가적인 정보를 자유롭게 텍스트 형태로 기술할 수 있도록 함 분야별로 양식을 정하여서 작성한 후, 해당 요소에 입력

2.6.2 XML 작성 예시

다음은 XML형식으로 작성된 파일 메타데이터의 작성 예시이다.

```

<Identifier> F_XXXXXX1 </Identifier>
<Creator> 홍길동
  <englishName> Gil-Dong Hong </englishName>
  <nameIdentifier nameIdentifierscheme = "과학기술인등록번호">123457
</nameIdentifier> </Creator>
<Title> 파일 메타데이터 정의
  <englishTitle> Definition of Metadata for Files </englishTitle>
  <additionalTitle titleType = "부제목"> 파일의 생산, 등록, 접근에 대한 메타정보의 정의
</additionalTitle>
</Title>
<Publisher> 한국과학기술정보연구원
  <englishName> Korea Institute of Science and Technology Information
</englishName>
</Publisher>
<PublicationYear> 20151021 </PublicationYear>
<Subject subjectScheme = "국가과학기술표준분류"> EE9999. 달리 분류되지 않는
정보/통신 </Subject>
<Dates datesType = "생성일"> 20150914 </Dates>
<Dates datesType = "제출일"> 20150916 </Dates>
<Dates datesType = "발행일"> 20151001 </Dates>
<Accesspolicy> 조건부
  <accessDate> 20150111to20160110 </accessDate>
  <accessCharge> 0 </accessCharge>
  <accessParty> KISTI 빅데이터센터 </accessParty>
  <accessCondition> 비상업적 목적으로만 활용 가능 </accessCondition>
</Accesspolicy>
<Project> 국가 과학데이터 거버넌스 체계 수립 사업 </Project>
<Collection> 국가 과학데이터 거버넌스 체계 수립 사업 </Collection>
<Item> 메타데이터 표준화 </Item>
<DataCoverage coverageType="수집기간"> 20150101to20151001 </DataCoverage>
<DataCoverage coverageType="수집지역"> 서울시 </DataCoverage>
<DataQualityLV> 3 </DataQualityLV>

```

```

<Contributor>
  <contributorType> 주관기관 </contributorType>
  <contributorName> 한국과학기술정보연구원
    <nameIdentifier nameIdentifierScheme = "사업자등록번호"> 1234567
</nameIdentifier>
  </contributorName>
</Contributor>
<Language> 한국어 </Language>
<ResourceType resourceTypeGeneral = "문서"> 보고서 </ResourceType>
<ReferenceID referenceIDType = "DOI">
http://dx.doi.org/XX.XXX1/XXXX.XXXX.XX.X.XXX </ReferenceID>
<Size> 2,500,000 </Size>
<Format> PPT (Power Point Template) </Format>
<Description descriptionType = "목차"> 1. 목적 및 범위
2. 메타데이터의 구조
3. 컬렉션 메타데이터
4. 아이템 메타데이터
5. 파일 메타데이터 </Description>
<CreatingApplication applicationVersion = "Standard 2007"> Microsoft Office Power
Point </CreatingApplication>
<DetailedDataInfo> Http://xxxxx.xxxxx.xxx/xxxxx.jpg </DetailedDataInfo>
<Right rightType = "CCL"> CC-BY-NC-SA </Right>
<RightStatement> Copyright, Korea Institute of Science and Technology Information,
2011 </RightStatement>
<Keyword> 메타데이터, 표준화, 메타정보, 과학데이터, 연구데이터 </Keyword>

```

표 21 파일 메타데이터를 XML 형태로 작성한 예시

3. 리포지토리 관리를 위한 메타데이터

3.1 목적과 범위

이 문서는 국가 과학데이터 센터에 리포지토리를 등록하여야 하는 각 분야 리포지토리 운영자와 이를 관리하기 위한 국가 과학데이터 센터 담당자, 그리고 리포지토리 메타정보를 활용하고자 하는 연구자 및 데이터 센터 운영자들을 위해 작성되었으며, 과학 분야의 리포지터리(repository) 메타데이터의 스키마와 요소의 정의, 작성방법 및 예시 등을 설명하고 있다.

3.2 리포지토리 메타데이터 스키마

- 필수 영역

리포지토리를 등록, 생성하기 위해 필수적으로 요구되는 메타데이터는 다음과 같다.

메타데이터 요소	정의	발생	생성주체	참조
RepositoryIdentifier	리포지토리를 식별하기 위한 ID	M, 1	자동	
RepositoryName	리포지토리의 명칭 (국문)	M, 1	등록자	
- englishName	리포지토리의 명칭 (영문)	M, 1	등록자	
- additionalName	리포지토리의 추가적인 명칭	O, 0-n	등록자	
-- nameType	(약어, 해당국가 언어명칭, 기타)로 유형 구분	M, 1	등록자	
RepositoryURL	해당 리포지토리의 URL	M, 1	등록자	
Type	해당 리포지토리의 유형(학술, 기관/단체, 기타)	M, 1	등록자	
RepositoryLanguage	리포지토리의 주 사용 언어	M, 1-n	등록자	
Subject	과학데이터 리포지토리의 주제	M, 1-n	등록자	
- subjectScheme	(과학기술표준분류,6T,국가전략기술,DDC분류체계,DFG분류체계,기타)으로 구분	M, 1	등록자	
EntryDate	리포지토리의 등록일자	M, 1	등록자	
LastDate	리포지토리에 대한 메타정보가 마지막으로 변경된 일자	M, 1	등록자	
InstitutionName	리포지토리를 생산, 펀딩, 운영하는 기관 명칭 (국문)	M, 1	등록자	
- institutionCountry	해당기관이 속한 국가	M, 1	등록자	
- institutionOriginalName	해당국가 언어의 공식 기관명	O, 1	등록자	
- institutionEnglishName	해당 기관의 영문 명칭	M, 1	등록자	
- responsibilityType	기관이 갖는 책임의 유형은 (생산, 펀딩, 운영) 중에서 택일	M, 1	등록자	
- responsibilityStartdate	리포지토리에 대한 책임이	M, 1	등록자	

	부여된 일자			
- responsibilityEnddate	리파지토리에 대한 책임이 종료되는 일자	O, 1	등록자	
- institutionContact	기관연락처(이메일 또는 메시지 입력 폼이 있는 URL)	M, 1	등록자	
DatabaseAccess	리파지토리에 대한 접근 유형(공개, 제한, 비공개)	M, 1	등록자	
- databaseAccessRestriction	리파지토리 접근에 대한 제한(제한일 경우, 요금제한/등록자제한 등으로 선택)	M, 1-n	등록자	
DataAccess	데이터 접근의 유형(공개, 임바고, 제한, 비공개)	M, 1	등록자	
- dataAccessRestriction	데이터 접근에 대한 제한(제한일 경우, 요금제한/등록자 제한 등 선택)	M, 1-n	등록자	
DataUpload	데이터 업로드 권한의 유형(공개, 제한, 비공개)	M, 1	등록자	
- dataUploadRestriction	데이터 업로드에 대한 제한(제한일 경우, 요금제한/등록자제한 등 선택)	M, 1-n	등록자	

표 22 리파지토리 메타데이터 필수영역 정의

● 선택 영역

리파지토리를 관리하기 위해 추가적으로 요구되는 메타데이터는 다음과 같다.

메타데이터 요소	정의	발생	생성주체	참조
Description	해당 리파지토리에 대한 설명	O, 1	등록자	
- Language	리파지토리에 대한 설명을 국문/영문으로 기입	M, 1	등록자	
RepositoryContact	해당 리파지토리의 담당자 연락처(이메일/URL)	O, 1	등록자	
Software	해당 리파지토리 운영을 위해 사용되는 S/W	O, 1	등록자	
API	API 의 URL	O, 1	등록자	
- APIType	(FTP, NetCDF, OAI-PMH, OpenDAP, REST, SOAP, SPARQL, SWORD, other)으로 유형 구분	M, 1	등록자	

3.3 요소 정의 및 작성 방법

요 소 명	RepositoryIdentifier
관련 요소	[None]
상세 설명	리파지토리를 식별하기 위한 ID
작성 방법	시스템에서 자동적으로 발급하여 부여함
작성 예시	R1234567

요 소 명	RepositoryName
관련 요소	[하위] englishName, additionalName
상세 설명	리파지토리의 명칭
작성 방법	해당 리파지토리의 공식 국문 명칭을 기입
작성 예시	듀크대학교 법학 리파지토리

요 소 명	englishName
관련 요소	[상위] RepositoryName
상세 설명	리파지토리의 명칭
작성 방법	해당 리파지토리의 공식 영문 명칭을 기입
작성 예시	Duke Law Scholarship Repository

요 소 명	additionalName
관련 요소	[상위] RepositoryName [하위] nameType
상세 설명	리파지토리의 추가적인 명칭
작성 방법	해당 리파지토리의 약어, 가칭 또는 다른 국가 언어 명칭을 기입
작성 예시	DLSR

요 소 명	nameType
관련 요소	[상위] additionalName
상세 설명	리파지토리의 추가적인 명칭의 유형
작성 방법	해당 리파지토리에 대한 추가적인 명칭의 유형을 정해진 값 내에서 선택하여 기입함 리파지토리를 추가적으로 설명하기 위해 많이 사용되어지는 약어, 가칭 또는 다른 국가 언어 명칭 중에 선택하여 기입함
작성 예시	약어

요 소 명	RepositoryURL
관련 요소	[None]
상세 설명	해당 리파지토리의 URL
작성 방법	사용자들이 리파지토리에 접근할 수 있도록 URL을 제공함
작성 예시	http://scholarship.law.duke.edu/

요 소 명	Type
관련 요소	[None]
상세 설명	리파지토리의 유형
작성 방법	해당 리파지토리의 유형을 (학술, 기관/단체, 기타) 중에서 선택하여 기입
작성 예시	학술

요 소 명	RepositoryLanguage
관련 요소	[None]
상세 설명	리파지토리의 주 사용 언어
작성 방법	해당 리파지토리에서 주로 사용되어지는 언어를 기입
작성 예시	English

요 소 명	Subject
관련 요소	[하위] subjectScheme
상세 설명	해당 리파지토리의 주제 분야
작성 방법	주제 스키마를 선택하고, 그 스키마에 따라서 주제 분야를 기입
작성 예시	SA0103. 법정책

요 소 명	subjectScheme
관련 요소	[상위] Subject
상세 설명	해당 리파지토리의 주제 스키마
작성 방법	주제 스키마는 (과학기술표준분류, 6T, 국가전략기술, DDC 분류체계, DFG 분류체계, 기타) 중에서 선택하여 입력 기타로 선택하였을 경우, 스키마를 새롭게 등록하여 사용할 수 있음
작성 예시	과학기술표준분류

요 소 명	EntryDate
관련 요소	[None]
상세 설명	리파지토리의 등록일자
작성 방법	해당 리파지토리를 시스템에 등록한 일자를 기입 시스템에 등록할 때 자동적으로 입력
작성 예시	2015년 09월 22일

요 소 명	LastDate
관련 요소	[None]
상세 설명	리파지토리의 업데이트 일자
작성 방법	리파지토리에 대한 메타정보가 마지막으로 변경된 일자를 기입 메타 데이터 변경 시 시스템에서 자동적으로 갱신
작성 예시	2015년 10월 01일

요 소 명	InstitutionName
관련 요소	[하위] institutionCountry, institutionOriginalName, institutionEnglishName, responsibilityType, responsibilityStartdate, responsibilityEnddate, institutionContact
상세 설명	리파지토리와 관련된 기관들의 명칭
작성 방법	기관의 책임유형에 따라 생산, 편딩, 운영 등과 관련된 기관들의 명칭을 기입함 국문 명칭을 기입함
작성 예시	듀크 법대

요 소 명	institutionCountry
관련 요소	[상위] InstitutionName
상세 설명	기관이 위치한 국가 명칭
작성 방법	해당 기관이 위치한 국가 명칭을 기입함 국가명은 검색하여 입력할 수 있도록 기능을 지원함
작성 예시	미국 (United States)

요 소 명	institutionOriginalName
관련 요소	[상위] InstitutionName
상세 설명	해당 기관의 공식 명칭
작성 방법	해당 기관을 설립한 국가의 언어로 공식명칭을 기입 국제기관의 경우, 기관에서 사용하는 주 언어로 표현된 공식 명칭을 기입
작성 예시	Duke Law School

요 소 명	institutionEnglishName
관련 요소	[상위] InstitutionName
상세 설명	해당 기관의 영어 명칭
작성 방법	해당 기관의 명칭을 영문으로 기입
작성 예시	Duke Law School

요 소 명	responsibilityType
관련 요소	[상위] InstitutionName
상세 설명	해당 기관의 책임 유형
작성 방법	해당 기관이 리파지토리에 대한 어떤 책임을 갖고 있는지 기입 (생산, 편당, 운영, 기타) 중에서 선택하여 값을 기입 기타의 경우, 자유롭게 텍스트 값을 추가 기입
작성 예시	운영

요 소 명	responsibilityStartdate
관련 요소	[상위] InstitutionName
상세 설명	해당 기관에게 책임이 부여된 일자
작성 방법	해당 기관에 리파지토리와 관련된 책임이 부여된 일자를 기입
작성 예시	1645년

요 소 명	responsibilityEnddate
관련 요소	[상위] InstitutionName
상세 설명	해당 기관의 책임이 만료되는 일자
작성 방법	리파지토리와 관련된 기관의 책임이 만료되는 일자를 기입 만료 예정일이 없으면, 기입하지 않음
작성 예시	

요 소 명	institutionContact
관련 요소	[상위] InstitutionName
상세 설명	기관 연락처
작성 방법	리파지토리와 관련된 책임사항을 문의할 수 있도록 담당자의 연락처를 기입 이메일 주소 또는 메시지를 남길 수 있는 입력 폼 주소(URL)를 기입
작성 예시	dukelaurepository@duke.edu

요 소 명	DatabaseAccess
관련 요소	[하위] databaseAccessRestriction
상세 설명	리파지토리에 대한 접근 유형
작성 방법	해당 리파지토리에 대한 접근 유형을 (공개, 제한, 비공개) 중에서 선택하여 입력
작성 예시	공개

요 소 명	databaseAccessRestriction
관련 요소	[상위] DatabaseAccess
상세 설명	리파지토리 접근에 대한 제한 유형
작성 방법	해당 리파지토리에 대한 접근을 제한할 경우, 요금제한, 사용자제한, 기타 등으로 선택하여 제한 유형을 입력
작성 예시	

요 소 명	DataAccess
관련 요소	[하위] dataAccessRestriction
상세 설명	데이터 접근의 유형
작성 방법	해당 리파지토리에 대한 접근 유형을 (공개, 임바고, 제한, 비공개) 중에서 선택하여 입력
작성 예시	제한

요 소 명	dataAccessRestriction
관련 요소	[상위] DataAccess
상세 설명	데이터 접근에 대한 제한 유형
작성 방법	해당 리파지토리에 대한 접근을 제한할 경우, 요금제한, 등록자 제한, 기타 등으로 제한조건을 입력
작성 예시	등록자 제한

요 소 명	DataUpload
관련 요소	[하위] dataUploadRestriction
상세 설명	데이터 업로드 권한의 유형
작성 방법	해당 리퍼지토리에 대한 데이터 업로드 권한 유형을 (공개, 제한, 비공개) 중에서 선택하여 입력
작성 예시	제한

요 소 명	dataUploadRestriction
관련 요소	[상위] DataUpload
상세 설명	데이터 업로드에 대한 제한 유형
작성 방법	해당 리퍼지토리에 대한 데이터 업로드를 제한할 경우, 요금제한, 등록자 제한, 기타 등 으로 제한조건을 입력
작성 예시	등록자 제한

요 소 명	Description
관련 요소	[하위] Language
상세 설명	해당 리퍼지토리에 대한 설명
작성 방법	해당 리퍼지토리에 대한 설명을 텍스트 형태로 자유롭게 기술
작성 예시	This is a subject based institutional repository providing access to the scholarly works produced by the Duke Law faculty. The interface is in English. This site replaces a previous EPrints repository.

요 소 명	Language
관련 요소	[상위] Description
상세 설명	리퍼지토리 설명을 기술한 언어명
작성 방법	리퍼지토리 설명을 무슨 언어로 기술하였는지 기입
작성 예시	English

요 소 명	RepositoryContact
관련 요소	[None]
상세 설명	해당 리파지토리의 담당자 연락처
작성 방법	리파지토리와 관련된 내용을 문의할 수 있도록 담당자의 연락처를 기입 이메일 주소 또는 메시지를 남길 수 있는 입력 폼 주소(URL)를 기입
작성 예시	Repository@dukelawschool.edu

요 소 명	Software
관련 요소	[None]
상세 설명	해당 리파지토리 운영을 위해 사용되는 S/W
작성 방법	리파지토리를 운영하기 위해 사용되는 소프트웨어의 명칭을 기입
작성 예시	Digital Commons

요 소 명	API
관련 요소	[하위] APIType
상세 설명	API 의 주소
작성 방법	해당 리파지토리의 API URL을 기입
작성 예시	http://scholarship.law.duke.edu/do/oai/

요 소 명	APIType
관련 요소	[상위] API
상세 설명	API 의 주소 유형
작성 방법	해당 리파지토리의 API 주소 유형을 (FTP, NetCDF, OAI-PMH, OpenDAP, REST, SOAP, SPARQL, SWORD, other) 중에서 선택하여 입력
작성 예시	OAI-PMH

3.4 XML 작성 예시

다음은 XML 형식으로 작성된 리파지토리 메타데이터의 작성 예시이다.

```
<RepositoryIdentifier> R1234567 </RepositoryIdentifier>
<RepositoryName> 듀크대학교 법학 리파지토리
  <englishName> Duke Law Scholarship Repository </englishName>
  <additionalName nameType = "약어"> DLSR </additionalName>
</RepositoryName>
<RepositoryURL> http://scholarship.law.duke.edu/ </RepositoryURL>
<Type> 학술 </Type>
<RepositoryLanguage> English </RepositoryLanguage>
<Subject subjectScheme = "과학기술표준분류"> SA0103. 법정책 </Subject>
<EntryDate> 20150922 </EntryDate>
<LastDate> 20151001 </LastDate>
<InstitutionName> 듀크 법대
  <institutionCountry> United States </institutionCountry>
  <institutionOriginalName> Duke Law School </institutionOriginalName>
  <institutionEnglishName> Duke Law School </institutionEnglishName>
  <responsibilityType> 운영 </responsibilityType>
  <responsibilityStartdate> 1645 </responsibilityStartdate>
  <responsibilityEnddate> </responsibilityEnddate>
  <institutionContact> dukelawrepository@duke.edu </institutionContact>
</InstitutionName>
<DatabaseAccess databaseAccessRestriction = " "> 공개 </DatabaseAccess>
<DataAccess dataAccessRestriction = "등록자 제한"> 제한 </DataAccess>
<DataUpload dataUploadRestriction = "등록자 제한"> 제한 </DataUpload>
<Description Language = "English"> This is a subject based institutional repository
providing access to the scholarly works produced by the Duke Law faculty. The
interface is in English. This site replaces a previous EPrints repository.
</Description>
<RepositoryContact> Repository@dukelawschool.edu </RepositoryContact>
<Software> Digital Commons </Software>
<API APIType = "OAI-PMH"> http://scholarship.law.duke.edu/do/oai/ </API>
```

표 24 리파지토리 메타데이터를 XML 형태로 작성한 예시

4. 메타데이터 전송

4.1 목적과 범위

이 문서는 ‘국가과학데이터센터’와 연계되는 리포지터리(repository) 운영자 및 개발자에게 메타데이터를 전송하는 방법을 설명하는데 있다. 또한 시스템 간 메타데이터 전송에 관심이 있는 사람에게도 유익하다. 메타데이터를 전송하는 방법은 여러 가지가 있지만 본 문서에서는 직접 전송(direct harvest)과 OAI-PMH 프로토콜을 이용한 전송방법만 설명한다. OAI-PMH 프로토콜을 이용한 전송방법의 경우 OAI-PMH 공식사이트의 [문서](#)를 번역 사용하였으며, 독자의 이해를 돕기 위해 일부 내용을 편집하였다.

4.2 직접 전송

작성예정입니다.

4.3 OAI-PMH 프로토콜을 이용한 전송

‘국가과학데이터센터(NSDC)’와 메타데이터를 연계하기 위해서는 연계대상 리포지터리에 Open Archives Initiative Protocol for Metadata Harvesting (이하, OAI-PMH)을 지원하는 기능이 구현되어 있어야 한다. NSDC에서 개발 보급되고 있는 리포지터리(Datanest)에는 센터와 메타데이터를 교환하기 위해서 기본기능으로 OAI-PMH가 구현되어 있다.

다음 내용은 OAI-PMH를 직접 구현하고자 하는 리포지터리 운영자나 개발자를 위해서 OAI-PMH 공식 문서를 번역, 보강한 내용이다.

OAI-PMH는 응용프로그램에 독립적인 시스템 상호운용성을 포장하는 프레임워크(application-independent interoperability framework)로서 기본적으로 메타데이터 수집(metadata harvesting)을 위한 것이다.

4.3.1 OAI-PMH 프레임워크

OAI-PMH 프레임워크에는 아래와 같이 두 개의 그룹(classes)이 필요하다.

- 데이터 제공자(Data Providers) 그룹은 메타데이터의 노출(exposing) 수단으로 OAI-PMH를 지원하는 시스템을 운영함
-
- 서비스 제공자(Service Providers)들은 OAI-PMH에 의해 수집된 메타데이터를 이용하여 가치가 부여된 서비스를 제공함

4.3.2 OAI-PMH 용어정리

OAI-PMH에서 사용되는 용어를 정리하면 다음과 같다.

- Harvester
 - OAI-PMH 요구를 발생시키는 클라이언트 응용프로그램
 - 서비스 제공자가 리포지터리(repository)로부터 메타데이터를 수집하기 위해 운영

- Repository

- 6개의 OAI-PMH 요청을 처리할 수 있는 네트워크로 접근 가능한 서버
- Harvester가 메타데이터를 수집할 수 있도록 데이터 제공자에 의해 운영
- OAI-PMH에서는 리포지터리의 다양한 환경을 인정한다. 다만 다음의 3개 엔티티(entity)를 구분하여 정의하고 있다.

- ✓ resource (자원)

자원은 객체(object, 혹은 stuff)로서 물리적(physical) 자원이거나 디지털 자원일 수 있으며, 리포지터리 내에 존재할 수도, 그렇지 않을 수도 있다. 다만, OAI-PMH에서 다루지 않는다.

- ✓ item (아이템)

하나의 아이템은 리포지터리의 구성요소이다. 아이템으로부터 메타데이터가 배포(dissemination)될 수 있다. 하나의 아이템은 개념적으로 하나의 컨테이너를 의미한다. 아이템은 하나의 자원에 대한 메타데이터를 다양한 형식으로 가지고 있거나 동적으로 생성할 수 있다. 각각의 메타데이터는 OAI-PMH를 통해 레코드들(records)로 수집될 수 있다. 각각의 아이템은 하나의 리포지터리 내에서 유일한 식별자(unique identifier)를 가지고 있다. 이 식별자는 아이템으로부터 메타데이터를 추출할 때 사용된다. 하나의 식별자는 하나의 아이템과 매핑이 되며, 하나의 아이템에 있는 여러 형식의 레코드들은 하나의 식별자를 공유한다.

- ✓ record (레코드)

특정한 메타데이터 형식을 따르는 메타데이터를 의미한다. 레코드는 특정형식의 메타데이터를 전송해 달라는 요청에 대한 응답으로 아이템으로부터 XML로 인코딩된 바이트스트림으로 전송된다.

- ✓ identifier (식별자)

식별자는 [URI \(Uniform Resource Identifier\)의 형식](#)을 따라야 한다. 식별자는 프로토콜에서 메타데이터를 요청하거나 제공할 때 사용된다. 응답의 경우, [ListIdentifiers](#)와 [ListRecords](#) 요청에 식별자들이 전달되며, 요청의 경우, 아이템으로부터 특정한 메타데이터 형식의 레코드를 요청하기 위해서 [GetRecord](#) 요청을 할 때 [metadataPrefix](#)와 함께 전

달된다.

이 식별자는 자원(resource)에 부여된 것이 아니고 아이템에 부여된 것이다. 따라서 자원으로의 접근이 가능하도록 제공되는 레코드에는 자원의 식별자가 함께 제공되어야 한다. OAI-PMH에서 의무적으로 제공해야 하는 [더블린 코어\(Dublin Core\) 메타데이터 형식](#)에서는 'identifier' 요소를 사용하여 식별자 값을 제공해야 한다.

4.3.3 Record

하나의 레코드는 [아이템 식별자](#)와 메타데이터 형식을 지정한 [metadataPrefix](#), 레코드의 [datestamp](#)로 식별된다. 레코드의 구성은 다음과 같다.

- header
 - 아이템의 식별자와 레코드의 선택적 수집을 위한 속성들(properties)로 구성
 - ✓ 아이템 식별자(unique identifier) : 리포지터리 내에서 아이템 식별자
 - ✓ datestamp : 선택적 수집을 위한 레코드의 생성, 수정, 삭제 일시
 - ✓ (zero or more) setSpec : 선택적 수집을 위한 아이템의 set 구성
 - ✓ status (attribute) : 조건적으로 사용될 수 있는 status 요소속성(attribute)으로서 그 요소속성의 값은 'deleted'로 정해져 있다. 이것은 아이템의 특정 메타데이터 형식이 삭제되었기 때문에 제공할 수 없음을 의미한다. 물론 리포지터리에서 deletions 기능을 제공해야 가능하다.
- metadata
 - 아이템으로부터 추출된 한 가지 형식의 메타데이터. 리포지터리는 한정어¹⁾(Qualifiers)를 포함하지 않은 최소한의 DC 형식 메타데이터를 제공해야 함.
 - 특정한 메타데이터 형식은 ListRecords와 GetRecord에 인자로 전달되는 metadataPrefix로 지정된다. ListMetadataFormats은 리포지터리에서 제공 가능한 모든 메타데이터 형식을 전달한다.
- about
 - metadata 부분에 대한 데이터를 XML 형식으로 표현하는 컨테이너로서 사용여부는 옵션이며 반복사용 가능 하다. 일반적으로 메타데이

1) [DCMES](#)의 활용에 있어 의미가 분명한 용어들(terms)과 인코딩 스킴(scheme)을 정의한 것을 의미한다.

터와 관련된 권한(rights)과 히스토리(provenance) 정보를 담고 있다.

- 권한의 경우, OAI-PMH로 전송되는 메타데이터의 사용에 관한 사용 조건이 주로 기술된다. 권한 정보를 기술하기 위한 특별한 태그를 권고하지 않는다.
- 히스토리의 경우, 제공되는 메타데이터가 수집되어 재배포 되는 경우 (수집이력이 있을 경우)에는 어떤 리포지터리로부터 수집되었는지? 언제 수집되었는지 정보가 사용권고 되고 있다.

```
<about>
<provenance xmlns="http://www.openarchives.org/OAI/2.0/provenance"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/provenance
    http://www.openarchives.org/OAI/2.0/provenance.xsd">

  <originDescription harvestDate="2002-02-02T14:10:02Z" altered="true">
    <baseURL>http://the.oa.org</baseURL>
    <identifier>oai:r2.org:klik001</identifier>
    <datestamp>2002-01-01</datestamp>

  <metadataNamespace>http://www.openarchives.org/OAI/2.0/oai_dc/</metadataNa
  mespace>
    <originDescription harvestDate="2002-01-01T11:10:01Z" altered="false">
      <baseURL>http://some.oa.org</baseURL>
      <identifier>oai:r2.org:klik001</identifier>
      <datestamp>2001-01-01</datestamp>

  <metadataNamespace>http://www.openarchives.org/OAI/2.0/oai_dc/</metadataNa
  mespace>
  </originDescription>
</originDescription>

</provenance>
</about>
```

표 25 About 요소의 사용 사례

(소스 출처: <http://www.openarchives.org/OAI/2.0/guidelines-provenance.htm>)

<표1>에는 originDescription 요소가 사용되었다. 수집된 메타데이터가 재배포될 수 있으며, 재배포되는 메타데이터는 originDescription 요소를 추가한다.

4.3.4 Deleted Record

OAI-PMH에서는 삭제된 레코드를 'deleted'로 표현한다. 리포지터리는 '[Identify](#)' 응답에 사용하는 'deletedRecord' 요소에 삭제된 레코드들을 지원하기 위한 3가지 수준 중 하나를 반드시 표기(declare)해야 한다.

- no
 - 리포지터리가 삭제에 대한 정보를 관리하지 않는다. 리포지터리는 레코드의 삭제 상태 정보를 제공하지 않는다.
- persistent
 - 시간의 제한 없이 삭제에 관한 정보를 유지한다. 리포지터리는 레코드의 삭제와 관련된 모든 히스토리 정보를 제공해야 한다.
- transient
 - 리포지터리는 삭제된 레코드에 대해서 지속적으로 정보를 제공할지 보장하지 않으며, 정보를 제공하지 않을 수도 있다.

<표2>에서는 '[Identify](#)' 요청에 리포지터리가 응답하는 내용의 일부이다. 마지막 줄은 'deletedRecord' 요소의 값을 'transient'로 지정하고 있다. 이것은 리포지터리가 삭제된 레코드에 대해서 지속적으로 정보를 제공할지 보장하지 않으며, 정보를 제공하지 않을 수도 있음을 의미한다.

```
<Identify>
  <repositoryName>Library of Congress Open Archive Initiative
    Repository 1</repositoryName>
  <baseURL>http://memory.loc.gov/cgi-bin/oai</baseURL>
  <protocolVersion>2.0</protocolVersion>
  <adminEmail>somebody@loc.gov</adminEmail>
  <adminEmail>anybody@loc.gov</adminEmail>
  <earliestDatestamp>1990-02-01T12:00:00Z</earliestDatestamp>
  <deletedRecord>transient</deletedRecord>  <-- 레코드 삭제정보에 관한 정책
```

표 26 레코드 삭제정보에 관한 리포지터리의 정책정보 제공 사례
(소스출처: <http://www.openarchives.org/OAI/openarchivesprotocol.html#Identify> 일부 소스를 사용)

리포지터리가 삭제 정보를 관리하지 않을 경우, 서비스 제공자는 삭제된 레코드를 수집할 수 있는 방법이 없다. 반면에 리포지터리가 삭제정보를 관리하는 경우, 삭제된 레코드의 'datestamp'는 레코드가 삭제된 날짜와 시간이어야 한다. 또한 삭제된 레코드를 'GetRecord'로 요청받을 경우 'header'의 요소속성으로 status="deleted"를 전송해야 하며, metadata와 about 부분은 전송하지 않아야 한다. 삭제된 레코드가 포함된 set과 날짜범위를 이용한 선택적 수집(selective harvesting)에 대한 응답에도 삭제된 레코드에 대한 header 정보를 포함해야 한다.

4.3.5 Set

Set은 서비스 제공자의 선택적 수집(selective harvesting)을 지원하기 위해 데이터 제공자가 아이템들을 그룹화 해 놓은 것이다. 리포지터리 운영자는 OAI-PMH를 통해 제공될 메타데이터를 Set으로 그룹화 할 수 있다.

Set으로 아이템들을 그룹화 했을 경우에는 ListIdentifiers와 ListRecords, GetRecord 요청에 대한 응답 내용에 Set 정보가 반드시 제공되어야 한다. set 정보는 item들의 헤더(header)에 포함된다.

set은 단순한 리스트 혹은 다 계층(multiple hierarchies) 형식으로 구성이 가능하다. 계층 구조의 set은 setSpec의 형식을 따른다. setSpec은 set들을 콜론(:) 으로 구분한 연속된 문자열이다. URI unreserved characters의 '2.3. Unreserved Characters' 부분에 선언되어 있는 문자는 모두 사용가능 하다. 문자열에는 set을 구분하는 콜론(:)을 사용할 수 없다. setSpec은 리포지터리 내에서 set을 식별할 수 있는 식별자이다.

2.3. Unreserved Characters

Data characters that are allowed in a URI but do not have a reserved purpose are called unreserved. These include upper and lower case letters, decimal digits, and a limited set of punctuation marks and symbols.

unreserved = alphanum | mark

mark = "-" | "_" | "." | "!" | "~" | "*" | "'" | "(" | ")"

표 27 setSpec을 만들기 위해 사용할 수 있는 문자.

(출처: <http://www.ietf.org/rfc/rfc2396.txt> Uniform Resource Identifiers (URI): Generic Syntax 문서의 일부 내용 발췌)

<표4>는 [ListSets](#) 요청에 대한 응답 내용을 보여준다. set 요소안에 하위위소 (node)로는 setSpec, setName이 오며 옵션으로 setDescription을 사용할 수 있다. setName은 사람이 읽을 수 있는 짧은 문자열을 값으로 갖는다. setDescription의 값으로는 XML로 인코딩된 set에 대한 데이터를 자유롭게 기술할 수 있다. 자세한 것은 [OAI-PMH 구현 가이드](#)를 참고하면 된다.

```

<ListSets ...>
  <set>
    <setSpec>A</setSpec>
    <setName>set A</setName>
  </set>
  <set>
    <setSpec>A:B</setSpec>
    <setName>set A:B</setName>
  </set>
  <set>
    <setSpec>B</setSpec>
    <setName>set B</setName>
  </set>
  <set>
    <setSpec>B:C</setSpec>
    <setName>set B:C</setName>
  </set>
  <set>
    <setSpec>B:D</setSpec>
    <setName>set B:D</setName>
  </set>
  <set>
    <setSpec>B:D:E</setSpec>
    <setName>set B:D:E</setName>
  </set>
</ListSets>

```

표 28 ListSets 요청에 대한 응답
 (소스출처: <http://www.openarchives.org/OAI/2.0/guidelines-repository.htm#MinimalImplementation-Sets>)

<표4>를 통해 다음과 같은 계층 구조로 되어 있음을 알 수 있다.

A
A:B
B
B:C
B:D
B:D:E

여기에서 A:B와 B는 전혀 다른 것으로 구별된다. 하지만 반복적인 B의 사용은 혼란을 주기 때문에 사용을 지양해야 한다.

<그림1>은 set의 포함관계 및 하나의 아이템이 여러 set에 포함될 수 있음을 도식화한 것이다. 그림에서와 같이 setB는 문자는 set의 명칭은 같지만 전혀 다른 것으로 취급된다. 따라서 setA:setB로 접근을 하면 item1의 정보를 얻을 수 있으나 setB로 접근하면 item4의 정보를 얻게 된다.

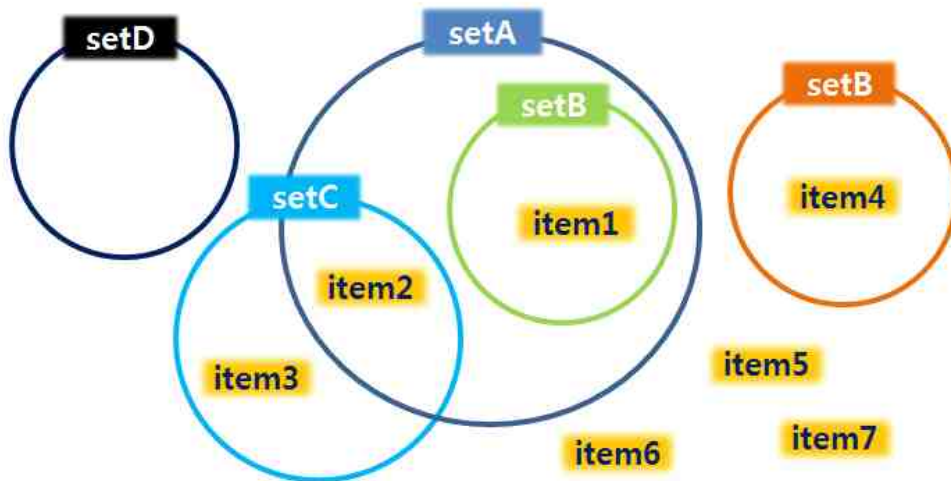


그림 8 set의 포함관계 및 하나의 아이템이 여러 set에 포함하는 개념도. 동일한 이름의 set이 전혀 다른 setSpec으로 정의 될 수 있음

만약에 아이템1에 대한 헤더가 다음과 같이 기술된다면

```
<header>  
  <identifier>item1</identifier>  
  <setSpec>A:B</setSpec>
```


...
</header>

아이템1은 set A와 A:B에 모두 속한다고 판단해야한다. 만약에 아이템2에 대한 헤더가 다음과 같이 기술된다면

```
<header>
  <identifier>item2</identifier>
  <setSpec>A</setSpec>
  <setSpec>B:D:E</setSpec>
  ...
</header>
```

아이템2는 set A, B, B:D 그리고 B:D:E에 속한다고 판단해야 한다. 이를 도식화 하면 <그림2>과 같다.

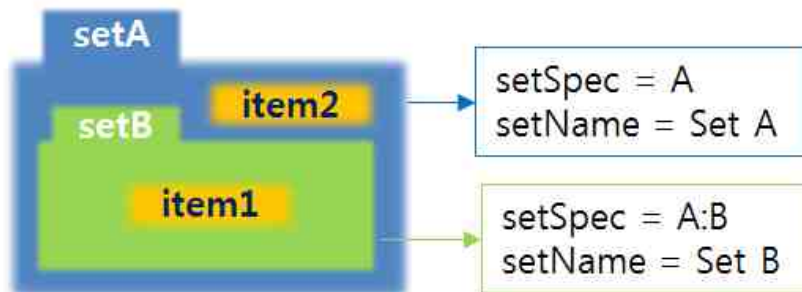


그림 9 setSpec과 setName

선택적 수집을 위해서 [ListRecords](#)과 [ListIdentifiers](#) 요청 인자(argument)에 set 인자가 사용될 수도 있다. set 인자의 값으로는 setSpec을 채워 넘겨준다.

Set 사용에 있어 다음의 다섯 가지 주요 사항을 고려해야 한다.

- 리포지터리가 sets을 지원할 경우 ListIdentifiers, ListRecords, GetRecord 요청에 set 구성정보를 제공해야 한다. 헤더에 아이템에 대한 setSpec 정보를 제공할 때 최소의 setSpec을 제공해야 한다. 예를 들어, <그림2>와 같은 경우에 item1에 대한 setSpec으로 'A:B'

만 제공하면 된다. setSpec으로 'A'를 제공할 필요가 없다. 'A:B'가 이미 'A'도 의미하기 때문이다.

- 하나의 아이템은 여러 set에 소속(조직)될 수 있다. 따라서 서로 다른 setSpec 인자(argument)로 메타데이터를 수집할 경우, 동일한 레코드가 수집될 수 있다.
- 아이템이 set에 꼭 소속될 필요는 없다. 따라서 반복적으로 모든 setSpec을 이용해서 ListRecords 요청을 보낸다 하더라도 리포지터리의 모든 레코드가 수집된다는 것을 보장할 수 없다. 결국, 모든 레코드를 수집하기 위해서는 setSpec 인자 없이 ListIdentifiers, ListRecords를 사용하는 것이다.
- 상위 set을 대상으로 레코드를 수집할 경우 하위(자손, descendant) set의 레코드도 모두 수집된다. <그림2>에서 setSpec 인자로 'A'를 전송할 경우, 아이템2와 setSpec 'A:B'의 아이템1이 모두 수집된다.
- 아이템을 가지고 있지 않은 set이 존재할 수 있다. <그림1>에서 setD가 그러한 예이다.

4.3.6 Selective Harvesting

- Selective Harvesting and Datestamps
 - 날짜에 기반 한 선택적 수집(datestamp-based selective harvesting)을 말한다. 특정 날짜를 기준으로 생성, 삭제, 수정된 레코드를 수집한다. 선택적으로 사용될 수 있는 인자(argument)로 'from'과 'until'이 ListIdentifiers, ListRecords 요청에서 사용된다. 가장 오래된 레코드를 수집하고 싶으면 from 인자를 최근 레코드까지 수집하고 싶으면 until 인자를 생략하면 된다. until의 값은 from의 값 보다 크거나 같아야 한다. 그렇지 않으면 리포지터리에서 'badArgument' 오류를 발생시킨다.
 - 초 단위까지의 범위 검색을 리포지터리가 지원하는지 여부는 Identify 요청을 통해 확인 할 수 있다. 요청과 응답에서 사용되는 datestamp 형식은 반드시 UTCdatetime을 따라야 한다. 리포지터리는 레코드의 datestamp를 반드시 업데이트 해야 한다. XML로 인코딩 되어 있는 metadata 부분에만 국한된 것이 아니며, 레코드의 메타데이터 형식 변화, 새로운 메타데이터 형식, 메타데이터 형식의 지원 중단과 같은 모든 상황 발생시 datestamp를 업데이트 해야 한다.

- Selective Harvesting and Sets
 - 선택적 수집을 지원하기 위해서 set 구성(set membership)이 ListIdentifiers, ListRecords 요청에서 사용 될 수 있다. set 인자에 setSpec 값이 사용된다. 상위 setSpec이 사용될 경우, 하위 set도 검색의 대상이 된다.

4.3.7 HTTP에 담긴 OAI-PMH 요청

HTTP GET 혹은 POST 방식으로 OAI-PMH 요청을 전송한다. POST 방식은 전송하는 인자들(arguments)의 길이에 제한이 없다. 리포지터리는 GET방식과 POST방식의 요청에 응답하기 위한 기본 URL(baseUrl)을 제공한다.

모든 요청은 'key=value' 형식을 갖는 'keyword arguments'로 구성되어 있다. 인자들의 순서는 의미가 없으며, 각 인자들은 '&' 기호로 구분된다. 요청은 최소 key의 값으로 'verb', value 값으로 'OAI-PMH 요청에서 정의한 하나의 하나의 값으로 구성되어야 한다.

- HTTP GET 방식으로 OAI-PMH 요청 보내기
 - 리포지터리 기본 URL 뒤에 '물음표(?)'를 붙이고 키워드 인자들을 '&'로 구분지어 연속 시킨다. 기본 URL이 'http://datacite.kr/oai-req' 라 했을 때 다음과 같이 요청을 생성할 수 있다.

```
http://datacite.kr/oai-req?  
verb=GetRecord&identifier=oai:arXiv.org:hep-th/9901001&metadataPr  
efix=oai_dc
```

- 물음표 앞부분 검은색 문자열은 OAI-PMH 요청을 받아들이는 리포지터리 서비스 URL을 의미한다. 연두색 문자열은 key를 의미하며, 파란색 부분은 value에 해당 된다. 각각의 키워드 인자들은 '&' 기호로 연결되어 있다. 위와 같은 URI에서는 특수문자가 인코딩 되어야 한다. 인코딩 후 문자열은 다음과 같다.

```

http://datacite.kr/oai-req?
verb=GetRecord&identifier=oai%3AarXiv.org%3Ahep-th%2F9901001&metadataPrefix
=oai_dc

```

- HTTP POST 방식으로 OAI-PMH 요청 보내기
 - 키워드 인자들을 메시지 몸체(message body)에 담아 전송한다. 요청의 Content-Type 값으로는 'application/x-www-form-urlencoded'을 갖도록 한다.

```

POST http://datacite.kr/oai-req
Content-Length: 82
Content-Type: application/x-www-form-urlencoded

verb=GetRecord&identifier=oai%3AarXiv.org%3Ahep-th/9901001&meta
dataPrefix=oai_dc

```

- 키워드 인자들의 값 중에 포함되어 있는 특수문자는 이스케이프 처리를 해주어야 한다.

Character	URI Role	Escape Sequence
/	Path Component Separator	%2F
?	Query Component Separator	%3F
#	Fragment Identifier	%23
=	Name/Value Separator	%3D
&	Argument Separator in Query Component	%26
:	Host Port Separator	%3A
;	Authority Namespace Separator	%3B
	Space Character	%20
%	Escape Indicator	%25
+	Escaped Space	%2B

표 32 키워드 인자값의 특수문자 인코딩 방법
(출처: <http://www.openarchives.org/OAI/openarchivesprotocol.html>)

4.3.8 HTTP 응답 형식

OAI-PMH 요청에 대한 응답은 HTTP 헤더 필드와 함께 HTTP 응답으로 이루어진다. 응답 형식(Content-Type)으로는 'text/xml'이 쓰인다. 리포지터리는

OAI-PMH를 구현한 서버로서 전송계층으로 HTTP를 사용한다. 따라서 [HTTP 상태코드 정의](#)를 따라야한다. 예를 들어

- 302 코드의 경우, 리포지터리의 일시적인 서비스 이동을 헤더의 Location 필드를 이용하여 전송할 수 있다.
- 503 코드의 경우, 헤더의 Retry-After 기간 필드를 이용하여 데이터 제공 서비스 보류기간을 전송할 수 있다.

<그림3>은 GetRecord 요청에 대한 정상적인 응답의 예시이다. 그림과 같이 모든 응답은 XML 형식으로 전달된다. 첫 번째 태그는 ‘<?xml version="1.0" encoding="UTF-8" ?>’으로 기술되어야 하며, 나머지 태그는 root 요소 ‘OAI-PMH’를(첫 번째 빨간 박스) 이용하여 기술한다. 이 루트 요소는 xmlns, xmlns:xsi, xsi:schemaLocation 요소 속성을 반드시 기술하여야 한다. <그림 3>에서 xsi:schemaLocation의 값(노란색 박스 부분)은 두 부분으로 구성되어 있다. 첫 번째 부분은 OAI-PMH의 이름공간 URI 이며, 두 번째 부분은 응답 결과의 정규화를 체크하기 위한 XML 스키마의 URL이다.

모든 응답에서 root 요소의 첫 번째 요소 두 개는 <그림3>에서 흰색 박스로 표현된 ‘responseDate’와 ‘request’이다. ‘responseDate’는 UTCdatetime 형식으로 된 응답 일시를 값으로 갖는다. ‘request’는 OAI-PMH요청을 받아들이는 기본 URL을 값으로 갖는다. 요청 시 인자로 넘어온 값을 key=value의 형식으로(요소 속성 형태) 갖는다. <그림3>에서는 verb, identifier, metadataPrefix 키를 요소 속성으로 기술하였다. 만약에 에러가 발생한 경우에는 기본 URL만을 값으로 갖으며, 요소 속성들을 기술하지 않는다. root 요소의 세 번째 요소는 다음과 같이 둘 중 하나가 올 수 있다.

- verb key와 같은 요소 : <그림3>에서는 GetRecord verb가 사용되었으므로 root 요소의 세 번째 요소로 GetRecord가 사용되었다.

```

<?xml version="1.0" encoding="UTF-8" ?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-05-01T19:20:30Z</responseDate>
  <request verb="GetRecord" identifier="oai:arXiv.org:hep-th/9901001"
    metadataPrefix="oai_dc">http://an.oa.org/OAI-script</request>
  <GetRecord>
    <record>
      ...
    </record>
  </GetRecord>
</OAI-PMH>

```

그림 10 GetRecord 요청에 대한 응답 예제

(소스 출처: <http://www.openarchives.org/OAI/openarchivesprotocol.html>)

- error 요소 : 에러가 발생 했을 때 사용되는 요소로서 <그림4>에서는 잘못된 verb가 사용되었을 때 error 요소의 사용 사례를 보여주고 있다. 예를 들어 존재하지 않는 verb로서 'nastyVerb'라는 것을 인자로 보낼 경우, <그림4>와 같은 결과를 받게 된다. 자세한 것은 다음 문서를 참조하기 바란다.

요청정보 :

<http://arXiv.org/oai2?verb=nastyVerb>

응답정보 :

```

<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-05-01T09:18:29Z</responseDate>
  <request>http://arXiv.org/oai2</request>
  <error code="badVerb">Illegal OAI verb</error>
</OAI-PMH>

```

그림 11 OAI-PMH에 존재하지 않는 verb 사용 시 전달되는 응답의 예시

(소스 출처: <http://www.openarchives.org/OAI/openarchivesprotocol.html>)

OAI-PMH 응답 결과를 검증하기 위한 스키마는 [여기](#)에서 확인 가능하다. <표 6>은 OAI-PMH verb와 관련된 오류코드와 오류내용을 보여주고 있다.

Error Codes	Description	Applicable Verbs
badArgument	The request includes illegal arguments, is missing required arguments, includes a repeated argument, or values for arguments have an illegal syntax.	all verbs
badResumptionToken	The value of the resumptionToken argument is invalid or expired.	ListIdentifiers ListRecords ListSets
badVerb	Value of the verb argument is not a legal OAI-PMH verb, the verb argument is missing, or the verb argument is repeated.	N/A
cannotDisseminateFormat	The metadata format identified by the value given for the metadataPrefix argument is not supported by the item or by the repository.	GetRecord ListIdentifiers ListRecords
idDoesNotExist	The value of the identifier argument is unknown or illegal in this repository.	GetRecord ListMetadata Formats
noRecordsMatch	The combination of the values of the from, until, set and metadataPrefix arguments results in an empty list.	ListIdentifiers ListRecords
noMetadataFormats	There are no metadata formats available for the specified item.	ListMetadata Formats
noSetHierarchy	The repository does not support sets.	ListSets ListIdentifiers ListRecords

표 33 OAI-PMH verb와 관련된 오류 코드와 오류 설명

(소스:<http://www.openarchives.org/OAI/openarchivesprotocol.html#ErrorConditions>)

4.4 metadataPrefix and Metadata Schema

[ListRecords](#)와 [GetRecord](#) 요청에 대한 응답은 [XML 네임스페이스 규격](#)을 준수하는 형태로 전송되어야 한다. 즉, root 요소는

- xmlns 속성(attribute)을 반드시 포함
- xsi:schemaLocation 속성을 반드시 포함

리포지터리의 상호 운용성(interoperability)을 위하여 [한정자](#) 없는 [Dublin Core\(DC\)](#)를 배포해야 한다. 따라서 OAI-PMH에서는 'oai_dc'라는 metadataPrefix와 DC를 위한 메타데이터 스키마의 URL, http://www.openarchives.org/OAI/2.0/oai_dc.xsd를 예약해 놓았다. 'all'이라는 metadataPrefix 또한 예약되어 있으므로 사용해서는 안 된다.

4.5 요청과 응답의 흐름 제어 (flow control)

아래의 요청들을 list requests라 한다.

- ListRecords는 records 리스트 리턴
- ListIdentifiers는 headers 리스트 리턴
- ListSets은 sets 리스트 리턴

반환되는 리스트의 크기가 클 경우, 리포지터리는 리스트 일부와 resumptionToken을 함께 반환한다. 데이터 수집 프로그램(harvester)은 반환된 토큰 인자(arguments) 을 사용하여 추가적인 요청(list request sequence라 함)을 수행한다. **최종적으로 반환되는 리스트에는 비어있는 (empty) resumptionToken 요소를 포함한다.** 이러한 용도 이외에 resumptionToken을 사용하는 것은 부정사용이며 오류를 반환해야 한다.

- 프로토콜에서는 resumptionToken의 형식을 별도로 지정하고 있지 않다.
- 추가적인 요청 시에 resumptionToken의 특수문자를 인코딩 해야 한다.
- 전체 리스트의 부분을 반환 할 경우, 불안정한 리스트의 의미가 정의 되어 있지 않다. 따라서 리스트 일부 전송의 기준을 추측해서는 안 된다.
- resumptionToken 요소는 다음과 같은 요소속성(attributes)을 옵션으로 사용한다.
-
- expirationDate -- resumptionToken이 유효한 날짜 시간 (UTCdatetime)
- completeListSize -- 반환되는 리스트의 총 개수. 연속 요청(list request sequence) 시 리포지터리 메타데이터의 변경(수정, 삭제)이 있을 수 있으므로 개수가 변경될 수 있다. 따라서 전체 반환 개수를 추측하는데 사용될 수 있음
- cursor - 현재까지 반환된 리스트 요소들의 개수. (0부터 시작함)

다음과 같은 상황을 가정하자. 175개 레코드를 전송해야 하는 ListRecords 요

청이 있다. 리포지터리는 100개의 레코드를 기준으로 응답을 한다. 이때 resumptionToken 사용 시나리오는 다음과 같다.

- 데이터 수집기(harvester)가 ListRecords 요청을 보냄
- 리포지터리는 resumptionToken의 요소 속성 completeListSize의 값으로 175, cursor 값으로 0을 100개의 레코드와 함께 반환
- 수집기는 resumptionToken을 파라미터 값으로 사용하여 ListRecords 요청을 다시 보냄
- 리포지터리는 75개 레코드를 반환. 이때 resumptionToken 요소 값 없이 cursor의 값은 100인 상태로 반환

resumptionToken을 사용하는 이유는 연속적인 요청으로 원하는 리스트를 순차적으로 반환받고자 하는데 있으며, 또 하나의 이유는 반환 값 전송 시에 발생할 수 있는 여러 가지 장애요소를 처리하고 요청을 재개하기 위함이다. resumptionToken은 다음과 같이 리포지터리의 두 가지 상황에서 처리될 수 있다.

- 리포지터리 내에 변화가 없을 경우 : 유효한 resumptionToken을 사용할 경우 동일한 반환 값을 제공
- 리포지터리 내에 변화가 있을 경우 : resumptionToken을 이용한 요청에 동일한 리스트 제공이 어려울 수 있으므로 resumptionToken을 이용한 재요청이 요구되지 않음. 대신에 반환되는 불완전한 리스트는 초기 요청에 사용된 datestamp 범위의 모든 레코드를 반환해야 함. 한편, 리포지터리는 badResumptionToken 에러를 발생하고 새로운 수집 요청을 유도하는 것이 바람직함.

4.6 프로토콜 요청과 응답

- GetRecord

- 하나의 메타데이터 레코드를 요청하는 verb(=token)
- 레코드가 요청되는 아이템의 식별자와 메타데이터 형식과 함께 사용됨
- 리포지터리로부터 혹은 지정된 아이템으로 부터 더 이상 유효하지 않은 metadataPrefix가 사용될 경우 status 요소속성 값으로 'deleted' 값을 가진 header가 반환될 수 있음. 이것은 리포지터리가 deletions 기능을 제공해야 함

- 필수 인자(Arguments)

- ✓ identifier

- ✓ metadataPrefix

- 오류와 예외 처리

- ✓ badArgument

- ✓ cannotDisseminatedFormat

- ✓ idDoseNotExist

- 예제

Dublin Core 메타데이터 형식으로 레코드를 요청하고 응답하는 예를 인코딩 없이 표현하면 다음과 같다.

- ✓ 요청

```
http://arXiv.org/oai2?
```

```
verb=GetRecord&identifier=oai:arXiv.org:cs/0112017&metadataPrefix=oai_dc
```

- ✓ 응답

```
<?xml version="1.0" encoding="UTF-8"?>
```

2) Deleted Record 부분 참조

```

<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-02-08T08:55:46Z</responseDate>
  <request verb="GetRecord" identifier="oai:arXiv.org:cs/0112017"
    metadataPrefix="oai_dc">http://arXiv.org/oai2</request>
  <GetRecord>
    <record>
      <header>
        <identifier>oai:arXiv.org:cs/0112017</identifier>
        <datestamp>2001-12-14</datestamp>
        <setSpec>cs</setSpec>
        <setSpec>math</setSpec>
      </header>
      <metadata>
        <oai_dc:dc
          xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
          xmlns:dc="http://purl.org/dc/elements/1.1/"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
            http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
          <dc:title>Using Structural Metadata to Localize Experience of
            Digital Content</dc:title>
          <dc:creator>Dushay, Naomi</dc:creator>
          <dc:subject>Digital Libraries</dc:subject>
          <dc:description>With the increasing technical sophistication of
            both information consumers and providers, there is
            increasing demand for more meaningful experiences of digital
            information. We present a framework that separates digital
            object experience, or rendering, from digital object storage
            and manipulation, so the rendering can be tailored to
            particular communities of users.
          </dc:description>
          <dc:description>Comment: 23 pages including 2 appendices,
            8 figures</dc:description>
          <dc:date>2001-12-14</dc:date>
        </oai_dc:dc>
      </metadata>
    </record>
  </GetRecord>
</OAI-PMH>

```

표 35 ListRecord 토큰으로 DC 형식의 레코드 요청과 정상적 응답
(소스 출처: <http://www.openarchives.org/OAI/openarchivesprotocol.html>)

Dublin Core 메타데이터 형식으로 레코드를 요청했는데 아이tem 식별자가 존재하지 않아 에러를 발생하는 경우의 예

✓ 요청

```
http://arXiv.org/oai2?  
verb=GetRecord&identifier=oai:arXiv.org:cs/0112017&metadataPrefix=oai_dc
```

✓ 응답

```
<?xml version="1.0" encoding="UTF-8"?>  
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/  
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">  
  <responseDate>2002-02-08T08:55:46Z</responseDate>  
  <request verb="GetRecord" identifier="oai:arXiv.org:quant-ph/0213001"  
    metadataPrefix="oai_dc">http://arXiv.org/oai2</request>  
  <error code="idDoesNotExist">No matching identifier in arXiv</error>  
</OAI-PMH>
```

표 37 ListRecord 토큰으로 DC 형식의 레코드 요청과 식별자 없음을 응답
(소스 출처: <http://www.openarchives.org/OAI/openarchivesprotocol.html>)

● Identify

- 리포지터리에 대한 정보를 요청할 경우 사용
- 필수 인자(Arguments)
 - ✓ 없음
- 오류와 예외 처리
 - ✓ badArgument
- 응답 형식
 - ✓ repositoryName
 - ✓ baseURL
 - ✓ protocolVersion
 - ✓ earliestDatestamp - 리포지터리에 기록되어 있는 datestamp의 최소 일시
 - ✓ deletedRecord - 삭제되는 레코드에 대한 정보를 리포지터리에서 어떻게

게 처리하는지 no, transient, persistent 값으로 표현. [deletion](#) 파트 참조

- ✓ granularity - 데이터 수집 시 사용될 수 있는 timestamps의 형식. [harvesting granularity](#) 참조
- ✓ adminEmail - 리포지터리 관리자 이메일. 하나 이상 기술될 수 있음
- 이하는 옵션으로 사용됨
- ✓ compression - HTTP 1.1을 기술한 RFC 2616의 섹션 14.11 파트의 content-Encoding header를 위해 정의되어 있는 값을 사용. *identity 인코딩은 포함되어 있으므로 포함되어서는 안 됨*
- ✓ description - 컬렉션 수준의 메타데이터를 포함하기 위해서 사용될 수 있음. 리포지터리를 기술하기 위해 사용됨. description 컨테이너에서 사용되는 요소들의 XML 스키마에 대한 URL이 함께 기술되어야 함. [OAI Identifier Format](#) 정보 등이 description 컨테이너에 기술될 수 있음

```
<description>
  <oai-identifier xmlns="http://www.openarchives.org/OAI/2.0/oai-identifier"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai-identifier
      http://www.openarchives.org/OAI/2.0/oai-identifier.xsd">
    <scheme>oai</scheme>
    <repositoryIdentifier>bespa.org</repositoryIdentifier>
    <delimiter>:</delimiter>
    <sampleIdentifier>oai:bespa.org:medi99-123</sampleIdentifier>
  </oai-identifier>
</description>
```

표 38 Identify 토큰의 응답 중 일부로서 description 컨테이너 활용 사례.

(소스 출처 :<http://www.openarchives.org/OAI/2.0/guidelines-oai-identifier.htm>)

domainname-word는 무조건 문자로 시작해야 한다. 따라서 콤마로 구분되는 각 문자열은 숫자로 시작할 수 없다. reserved 문자들은 local-identifier 내에서 의미가 없으므로 인코딩 될 필요가 없다. reserved와 unreserved에 포함되지 않는 모든 문자는 이스케이프 처리 되어야 한다. oai-identifier는 이스케이프 처리된 것을 해제해서는 안 된다. Hex 숫자는 A에서 F까지, 모두 대문자로 인코딩 되어야 한다.

```

oai-identifier = scheme ":" namespace-identifier ":" local-identifier

scheme = "oai"

namespace-identifier = domainname-word "." domainname
domainname = domainname-word [ "." domainname ]
domainname-word = alpha *( alphanum | "-" )

local-identifier = 1*uric

```

```

uric          = reserved | unreserved | escaped
reserved     = ";" | "/" | "?" | ":" | "@" | "&" | "=" | "+" | "$" | ","
unreserved   = alphanum | mark
mark         = "-" | "_" | "." | "!" | "~" | "*" | "." | "(" | ")"

```

```

escaped      = "%" hex hex
hex          = digit | "A" | "B" | "C" | "D" | "E" | "F"

```

domainname-word는 무조건 문자로 시작해야 함
따라서, 콤마로 구분되는 각 문자열은 숫자로 시작할 수 없음

reserved 문자들은 local-identifier 내에서 의미가 없으므로
인코딩 될 필요가 없음.

reserved와 unreserved에 포함되지 않는 모든 문자는 이스케이프 처리 되어야 함

oai-identifier는 이스케이프 처리된 것을 해제해서는 안됨

Hex 숫자는 A에서 F까지, 모두 대문자로 인코딩 되어야 함

그림 12 OAI Identifier 형식과 인코딩 규칙

- 예제

- ✓ 요청

```
http://memory.loc.gov/cgi-bin/oai?verb=Identify
```

표 39 Identify 토큰의 응답 중 일부로서 description 컨테이너 활용 사례.
(소스 출처 :<http://www.openarchives.org/OAI/2.0/guidelines-oai-identifier.htm>)

- ✓ 응답

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2002-02-08T12:00:01Z</responseDate>
<request verb="Identify">http://memory.loc.gov/cgi-bin/oai</request>
<Identify>
  <repositoryName>Library of Congress Open Archive Initiative
    Repository 1</repositoryName>
  <baseURL>http://memory.loc.gov/cgi-bin/oai</baseURL>
  <protocolVersion>2.0</protocolVersion>
  <adminEmail>somebody@loc.gov</adminEmail>
  <adminEmail>anybody@loc.gov</adminEmail>
  <earliestDatestamp>1990-02-01T12:00:00Z</earliestDatestamp>
  <deletedRecord>transient</deletedRecord>
  <granularity>YYYY-MM-DDThh:mm:ssZ</granularity>
  <compression>deflate</compression>
  <description>
    <oai-identifier
      xmlns="http://www.openarchives.org/OAI/2.0/oai-identifier"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation=
        "http://www.openarchives.org/OAI/2.0/oai-identifier
        http://www.openarchives.org/OAI/2.0/oai-identifier.xsd">
      <scheme>oai</scheme>
      <repositoryIdentifier>lcoa1.loc.gov</repositoryIdentifier>
      <delimiter>:</delimiter>
      <sampleIdentifier>oai:lcoa1.loc.gov:loc.music/musdi.002</sampleIdentifier>
    </oai-identifier>
  </description>
  <description>
    <eprints
      xmlns="http://www.openarchives.org/OAI/1.1/eprints"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.openarchives.org/OAI/1.1/eprints
      http://www.openarchives.org/OAI/1.1/eprints.xsd">
      <content>
        <URL>http://memory.loc.gov/ammem/oamh/lcoa1_content.html</URL>
        <text>Selected collections from American Memory at the Library
          of Congress</text>
      </content>
      <metadataPolicy/>
      <dataPolicy/>
    </eprints>
  </description>

```



```

<description>
  <friends
    xmlns="http://www.openarchives.org/OAI/2.0/friends/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/friends/
      http://www.openarchives.org/OAI/2.0/friends.xsd">
    <baseURL>http://oai.east.org/foo/</baseURL>
    <baseURL>http://oai.hq.org/bar/</baseURL>
    <baseURL>http://oai.south.org/repo.cgi</baseURL>
  </friends>
</description>
</Identify>
</OAI-PMH>

```

표 40 Identify 토큰에 대한 응답으로서 3개의 description 컨테이너 사용 예
 (소스 출처: <http://www.openarchives.org/OAI/openarchivesprotocol.html>)

● ListIdentifiers

- 레코드 정보를 반환받는 ListRecords와 달리 headers 정보만을 반환 받음
- set 구성이나 datestamp를 이용하여 선택적 수집 가능
- 전달되는 인자와 매핑되는 레코드가 삭제되었을 경우, 리포지터리의 deletions 지원 여부에 따라, header가 'deleted' 값을 갖는 status 요소속성을 가질 수 있음
- 필수 인자 (required arguments)
 - ✓ metadataPrefix
- 선택 인자 (optional arguments)
 - ✓ from
 - ✓ until
 - ✓ set
- 배타 인자 (exclusive arguments)
 - ✓ resumptionToken
- 오류와 예외 처리
 - ✓ badArgument

- ✓ badResumptionToken - 잘못되거나 만료된 resumptionToken
- ✓ cannotDisseminateFormat - 리포지터리에서 요청된 메타데이터 형식을 지원하지 않음
- ✓ noRecordsMatch - 조건에 맞는 레코드가 없음
- ✓ noSetHierarchy - 리포지터리가 sets을 지원하지 않음

- 예제

Dublin Core 메타데이터 형식으로 레코드를 요청하고 응답하는 예를 인코딩 없이 표현하면 다음과 같다.

- ✓ 요청

```

http://an.oa.org/OAI-script?
verb=ListIdentifiers&from=1998-01-15&metadataPrefix=oldArXiv&set=physics:hep

```

- ✓ 응답

```

<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-06-01T19:20:30Z</responseDate>
  <request verb="ListIdentifiers" from="1998-01-15"
    metadataPrefix="oldarXiv"
    set="physics:hep">http://an.oa.org/OAI-script</request>
  <ListIdentifiers>
    <header //..... 1번
      <identifier>oai:arXiv.org:hep-th/9801001</identifier>
      <timestamp>1999-02-23</timestamp>
      <setSpec>physics:hep</setSpec>
    </header>
    <header //..... 2번
      <identifier>oai:arXiv.org:hep-th/9801002</identifier>
      <timestamp>1999-03-20</timestamp>

```

```

<setSpec>physic:hep</setSpec>
<setSpec>physic:exp</setSpec>
</header>
<header> //..... 3번
<identifier>oai:arXiv.org:hep-th/9801005</identifier>
<timestamp>2000-01-18</timestamp>
<setSpec>physic:hep</setSpec>
</header>
<header status="deleted"> //..... 4번
<identifier>oai:arXiv.org:hep-th/9801010</identifier>
<timestamp>1999-02-23</timestamp>
<setSpec>physic:hep</setSpec>
<setSpec>math</setSpec>
</header>
<resumptionToken expirationDate="2002-06-01T23:20:00Z"
completeListSize="6"
cursor="0">xxx45abtttz</resumptionToken> //..... 5번
</ListIdentifiers>
</OAI-PMH>

```

표 42 ListIdentifiers 토큰에 대한 응답으로서 전달할 헤더는 총 6개로 구성되어 있으며 그중 4개의 헤더만 반환된 예
(소스 출처: <http://www.openarchives.org/OAI/openarchivesprotocol.html>)

<표11>-4번은 삭제된 레코드의 정보를 헤더의 요소속성으로 알려준다. 5번 resumptionToken은 completeListSize 요소속성으로 전체 리스트가 6개의 헤더로 구성되어 있음을 알려주고, cursor 요소속성을 통해 첫 번째 헤더(0번 부터 시작)부터 전송되었음을 알려준다. expirationDate을 통해 전달된 resumptionToken의 유효기간(시간)을 알려주고 있다. 본 예제에서는 2002년 6월 1일 23시20분 까지 유효함을 의미한다. 'xxx45abtttz'이 resumptionToken의 값으로 사용되었다. 아래에서는 전송된 resumptionToken을 가지고 요청하는 예를 보여준다.

✓ 요청

```

http://an.oa.org/OAI-script?
verb=ListIdentifiers&resumptionToken=xxx45abtttz

```

✓ 응답

```

<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
  http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-06-01T19:30:00Z</responseDate>
  <request verb="ListIdentifiers"

resumptionToken="xxx45abttyz">http://an.oa.org/OAI-script</request>
  <ListIdentifiers>
    <header> //..... 6번
      <identifier>oai:arXiv.org:hep-th/9801020</identifier>
      <datestamp>1999-02-23</datestamp>
      <setSpec>physic:hep</setSpec>
    </header>
    <header> //..... 7번
      <identifier>oai:arXiv.org:hep-th/9801060</identifier>
      <datestamp>1999-02-23</datestamp>
      <setSpec>physic:hep</setSpec>
    </header>
    <resumptionToken completeListSize="6" cursor="4"/> //..... 8번
  </ListIdentifiers>
</OAI-PMH>

```

표 44 ListIdentifiers 토큰에 resumptionToken을 전송하여 응답받은 예
(소스 출처: <http://www.openarchives.org/OAI/openarchivesprotocol.html>)

<표12>-6번, 7번은 resumptionToken에 의해 연속으로 전달된 헤더 값을 나타낸다. 헤더는 0부터 시작하기 때문에 8번의 cursor="4"는 전달된 첫 번째 헤더가 5번째 헤더임을 의미한다. 한편, 추가적인 resumptionToken 값이 전달되지 않았기 때문에 모든 리스트가 전달된 것으로 판단한다.

```

<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2002-02-08T14:27:19Z</responseDate>
<request verb="ListIdentifiers" metadataPrefix="olac"
from="2001-01-01" until="2001-01-01"
set="Perseus:collection:PersInfo">
http://www.perseus.tufts.edu/cgi-bin/pdataproduct</request>
<error code="noRecordsMatch"/> //..... 9번
</OAI-PMH>

```

표 45 요청 조건에 해당되는 레코드가 없을 경우, 헤더 없이 error 태그(tag)가 반환된 예 (소스 출처: <http://www.openarchives.org/OAI/openarchivesprotocol.html>)

<표13>-9번은 요청 조건에 해당되는 레코드가 없을 경우, 헤더 없이 error 태그(tag)가 반환된 예를 보여준다.

● ListMetadataFormats

- 리포지터리가 제공 가능한 메타데이터 형식을 요청하는 토큰(=verb)
- identifier 인자를 보낼 경우, 해당 아이템이 제공가능한 메타데이터 형식을 리턴함
- 필수 인자 (required arguments)
 - ✓ 없음
- 선택 인자 (optional arguments)
 - ✓ identifier
- 배타 인자 (exclusive arguments)
 - ✓ 없음
- 오류와 예외 처리
 - ✓ badArgument
 - ✓ idDoesNotExist
 - ✓ noMetadataFormats - 선택 인자에 해당하는 아이템이 제공하는 메타데이터 형식이 없을 경우

- 예제
식별자 'oai:perseus.tufts.edu:Perseus:text:1999.02.0119'에 해당하는
아이템의 제공 가능한 메타데이터 형식을 요청한다.

✓ 요청

```
http://www.perseus.tufts.edu/cgi-bin/pdatapro?
verb=ListMetadataFormats&identifier=oai:perseus.tufts.edu:Perseus:text:
1999.02.0119
```

✓ 응답

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
  http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-02-08T14:27:19Z</responseDate>
  <request verb="ListMetadataFormats"
    identifier="oai:perseus.tufts.edu:Perseus:text:1999.02.0119">
    http://www.perseus.tufts.edu/cgi-bin/pdatapro</request>
  <ListMetadataFormats>
    <metadataFormat>
      <metadataPrefix>oai_dc</metadataPrefix>
      <schema>http://www.openarchives.org/OAI/2.0/oai_dc.xsd
      </schema>
      <metadataNamespace>http://www.openarchives.org/OAI/2.0/oai_dc/
      </metadataNamespace>
    </metadataFormat>
    <metadataFormat>
      <metadataPrefix>olac</metadataPrefix>
      <schema>http://www.language-archives.org/OLAC/olac-0.2.xsd</schema>
      <metadataNamespace>http://www.language-archives.org/OLAC/0.2/
      </metadataNamespace>
    </metadataFormat>
    <metadataFormat>
      <metadataPrefix>perseus</metadataPrefix>
      <schema>http://www.perseus.tufts.edu/persmeta.xsd</schema>
      <metadataNamespace>http://www.perseus.tufts.edu/persmeta.dtd
```

```
</metadataNamespace>
</metadataFormat>
</ListMetadataFormats>
</OAI-PMH>
```

표 47 특정 아이템이 oai_dc, olac, perseus 3가지 형식으로 제공 가능함을 보여주는 반환 값

(소스 출처: <http://www.openarchives.org/OAI/openarchivesprotocol.html>)

✓ 요청

```
http://memory.loc.gov/cgi-bin/oai?
verb=ListMetadataFormats
```

✓ 응답

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-06-08T15:19:13Z</responseDate>
  <request verb="ListMetadataFormats">
    http://memory.loc.gov/cgi-bin/oai</request>
  <ListMetadataFormats>
    <metadataFormat>
      <metadataPrefix>oai_dc</metadataPrefix>
      <schema>http://www.openarchives.org/OAI/2.0/oai_dc.xsd</schema>
      <metadataNamespace>http://www.openarchives.org/OAI/2.0/oai_dc/
        </metadataNamespace>
    </metadataFormat>
    <metadataFormat>
      <metadataPrefix>oai_marc</metadataPrefix>
      <schema>http://www.openarchives.org/OAI/1.1/oai_marc.xsd</schema>
      <metadataNamespace>http://www.openarchives.org/OAI/1.1/oai_marc
        </metadataNamespace>
    </metadataFormat>
  </ListMetadataFormats>
</OAI-PMH>
```

표 49 리포지터리가 제공할 수 있는 모든 메타데이터 형식을 반환하는 예

(소스 출처: <http://www.openarchives.org/OAI/openarchivesprotocol.html>)

<표15>는 리포지터리 레벨에서 제공 가능한 메타데이터 형식을 반환한 예이다. 하지만, 리포지터리의 모든 아이템이 반환된 형식을 모두 제공할 수 있음을 의미하지 않는다.

```

<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-06-08T15:19:13Z</responseDate>
  <request verb="ListMetadataFormats"
    identifier="oai:lcoa1.loc.gov:loc.rbc/rbpe.00000111">
    http://memory.loc.gov/cgi-bin/oai</request> //..... 1번
  <error code="idDoesNotExist">oai:lcoa1.loc.gov:loc.rbc/rbpe.00000111 has the
    structure of a valid LOC identifier, but it maps to no known
    item</error> //..... 2번
</OAI-PMH>

```

표 50 요청된 식별자가 리포지터리에 존재하지 않음을 알리는 반환 값 예시
(소스 출처: <http://www.openarchives.org/OAI/openarchivesprotocol.html>)

<표16>-1번은 요청정보를 보여준다. ‘ListMetadataFormats’ 토큰을 사용하였으며, 선택적으로 사용 가능한 identifier 인자를 사용하였다. <표16>-2번을 통해 요청된 식별자가 LOC 식별자 구조로는 유효하나 리포지터리에 해당하는 아이템이 존재하지 않음을 알 수 있다.

- ListRecords

- 리포지터리로부터 레코드 리스트를 조회
- set 인자와 datestamp를 이용한 선택적 레코드 수집이 가능
- 리포지터리의 deletions 지원 여부에 따라서 헤더가 ‘deleted’ 값을 갖는 status 요소속성을 가질 수 있음. 이 경우에는 메타데이터 정보는 없음
- 필수 인자 (required arguments)

- ✓ metadataPrefix
 - 선택 인자 (optional arguments)
 - ✓ from, until, set
 - 배타 인자 (exclusive arguments)
 - ✓ resumptionToken
 - 오류와 예외 처리
 - ✓ badArgument
 - ✓ badResumptionToken - resumptionToken 이 잘못 되었거나 만료 되었음
 - ✓ cannotDisseminateFormat - 요청된 metadataPrefix 지원하지 않음
 - ✓ noRecordsMatch
 - ✓ noSetHierarchy - 리포지터리가 sets을 지원하지 않음
 - 예제

메타데이터 형식이 'oai_rfc1807'이며, '1998년 1월 15일'부터 추가되거나 수정된 아이템을 검색한다. 아이템들은 'physics'의 하위 set으로 'hep'에 포함되어 있다. 이하에서는 이러한 상황을 가정하여 작성된 예시이다.

✓ 요청

```
http://an.oa.org/OAI-script?
verb=ListRecords&from=1998-01-15&set=physics:hep&metadataPrefix=oai_rfc1807
```

✓ 응답

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
  http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
```

```

<responseDate>2002-06-01T19:20:30Z</responseDate>
<request verb="ListRecords" from="1998-01-15"
  set="physics:hep"
  metadataPrefix="oai_rfc1807">
  http://an.oa.org/OAI-script</request> //..... 1번
<ListRecords>
<record> //..... 2번
  <header> //..... 3번
    <identifier>oai:arXiv.org:hep-th/9901001</identifier>
    <timestamp>1999-12-25</timestamp>
    <setSpec>physics:hep</setSpec>
    <setSpec>math</setSpec>
  </header>
  <metadata> //..... 4번
    <rfc1807 xmlns=
      "http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1807.txt"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation=
        "http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1807.txt
        http://www.openarchives.org/OAI/1.1/rfc1807.xsd">
      <bib-version>v2</bib-version>
      <id>hep-th/9901001</id>
      <entry>January 1, 1999</entry>
      <title>Investigations of Radioactivity</title>
      <author>Ernest Rutherford</author>
      <date>March 30, 1999</date>
    </rfc1807>
  </metadata>
  <about> //..... 5번
    <oai_dc:dc
      xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
      http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
      <dc:publisher>Los Alamos arXiv</dc:publisher>
      <dc:rights>Metadata may be used without restrictions as long as
        the oai identifier remains attached to it.</dc:rights>
    </oai_dc:dc>
  <provenance //..... 6번

```

```

xmlns="http://www.openarchives.org/OAI/2.0/provenance"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/provenance
http://www.openarchives.org/OAI/2.0/provenance.xsd">
<originDescription harvestDate="2002-01-01T11:10:01Z" altered="true">
<baseURL>http://some.oa.org</baseURL>
<identifier>oai:r2.org:klik001</identifier>
<timestamp>2001-01-01</timestamp>
<metadataNamespace>http://www.openarchives.org/OAI/2.0/oai_dc
/</metadataNamespace>
</originDescription>
</provenance>
</about>
</record>
<record> //..... 7번
<header status="deleted"> //..... 8번
<identifier>oai:arXiv.org:hep-th/9901007</identifier>
<timestamp>1999-12-21</timestamp>
</header>
</record>
</ListRecords>
</OAI-PMH>

```

표 52 ListRecords 요청에 대한 응답으로서 두 개의 레코드 정보가 반환된 예 (소스 출처: <http://www.openarchives.org/OAI/openarchivesprotocol.html> 소스를 병합하여 하나로 표기 하였음)

<표17>-1번은 요청정보를 보여준다. 2번과 7번은 레코드 정보를 담고 있는 컨테이너 이다. 2번은 header, metadata, about 컨테이너로 구성되어 있다. 6번은 provenance 컨테이너를 이용해 출처정보를 제공하고 있다. 8번은 두 번째 레코드가 삭제되어 있음을 알려준다.

```

<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">

```

```

<responseDate>2002-06-01T19:20:30Z</responseDate>
<request verb="ListRecords" from="2002-06-01T02:00:00Z" //..... 1번
    until="2002-06-01T03:020:00Z"
    metadataPrefix="oai_marc">
    http://memory.loc.gov/cgi-bin/oai</request>
<error code="badArgument"/> //..... 2번
</OAI-PMH>

```

표 53 ListRecords 요청에서 from과 until 인자 값의 수준을 리포지터리가 지원하지 않는 경우의 예 (소스 출처: <http://www.openarchives.org/OAI/openarchivesprotocol.html>)

<표18>은 ListRecords 요청에서 from과 until 인자 값의 수준을 리포지터리가 지원하지 않는 경우의 예를 보여준다. 1번에서 사용된 from과 until의 인자값 수준(granularity)을 리포지터리가 지원하지 않기 때문에 2번과 같이 error 요소와 code 요소속성을 이용하여 'badArgument' 값을 반환하고 있다.

- ListSets

- 리포지터리의 set 구조를 검색하기 위해 사용하는 토큰으로서 레코드의 선택적 수집 시 유용함
- 필수 인자 (required arguments)
 - ✓ 없음
- 선택 인자 (optional arguments)
 - ✓ 없음
- 배타 인자 (exclusive arguments)
 - ✓ resumptionToken
- 오류와 예외 처리
 - ✓ badArgument
 - ✓ badResumptionToken
 - ✓ noSetHierarchy
- 예제

✓ 요청

```
http://an.oa.org/OAI-script?verb=ListSets
```

✓ 응답

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
  http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-08-11T07:21:33Z</responseDate>
  <request verb="ListSets">http://an.oa.org/OAI-script</request>
  <ListSets>
    <set> //..... 1번
      <setSpec>music</setSpec>
      <setName>Music collection</setName>
    </set>
    <set> //..... 2번
      <setSpec>music:(muzak)</setSpec>
      <setName>Muzak collection</setName>
    </set>
    <set>
      <setSpec>music:(elec)</setSpec>
      <setName>Electronic Music Collection</setName>
      <setDescription> //..... 3번
        <oai_dc:dc
          xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
          xmlns:dc="http://purl.org/dc/elements/1.1/"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
          http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
            <dc:description>This set contains metadata describing
              electronic music recordings made during the 1950ies
            </dc:description>
          </oai_dc:dc>
        </setDescription>
      </set>
    <set> //..... 4번
      <setSpec>video</setSpec>
```

```

    <setName>Video Collection</setName>
  </set>
</ListSets>
</OAI-PMH>

```

표 55 ListSets 요청에 대한 응답으로서 리포지터리의 set 정보 반환 예
(소스 출처: <http://www.openarchives.org/OAI/openarchivesprotocol.html>)

<표19>는 ListSets 요청에 대한 응답으로서 리포지터리의 set 정보를 반환한 예를 보여준다. 예제의 1번 ~ 4번을 통해 총 4개의 set으로 구성되어 있음을 알 수 있다. 3번의 경우에는 setDescription을 사용하여 해당 set에 대한 정보를 제공하고 있다.

Error Codes	Description	Applicable Verbs
badArgument	The request includes illegal arguments, is missing required arguments, includes a repeated argument, or values for arguments have an illegal syntax.	all verbs
badResumptionToken	The value of the resumptionToken argument is invalid or expired.	ListIdentifiers ListRecords ListSets
badVerb	Value of the verb argument is not a legal OAI-PMH verb, the verb argument is missing, or the verb argument is repeated.	N/A
cannotDisseminateFormat	The metadata format identified by the value given for the metadataPrefix argument is not supported by the item or by the repository.	GetRecord ListIdentifiers ListRecords
idDoesNotExist	The value of the identifier argument is unknown or illegal in this repository.	GetRecord ListMetadataFormats
noRecordsMatch	The combination of the values of the from, until, set and metadataPrefix arguments results in an empty list.	ListIdentifiers ListRecords
noMetadataFormats	There are no metadata formats available for the specified item.	ListMetadataFormats
noSetHierarchy	The repository does not support sets.	ListSets ListIdentifiers ListRecords

표 56 OAI-PMH verb와 관련된 오류 코드와 오류 설명
(소스: <http://www.openarchives.org/OAI/openarchivesprotocol.html#ErrorConditions>)

5. 참고자료

- [1] Metadata <http://ands.org.au/guides/metadata-awareness.html>
- [2] 메타데이터 기초 (Metadata Basics)
<http://dublincore.go.kr/metadata-basics>
- [3] Metadata <https://en.wikipedia.org/wiki/Metadata>
- [4] 메타데이터 <http://www.archives.go.kr/next/manager/metaData.do>
- [5] Understanding Metadata
<http://www.niso.org/publications/press/UnderstandingMetadata.pdf>
- [6] What is metadata? (Indiana University) <https://kb.iu.edu/d/aopm>
- [7] Metadata Basics <http://dublincore.org/metadata-basics/>
- [8] What is Metadata? - YouTube
<https://www.youtube.com/watch?v=96ncNPpohjo>
- [9] What is Metadata? - YouTube
<https://www.youtube.com/watch?v=HXAstVP3-y0>
- [10] Metadata standards
https://en.wikipedia.org/wiki/Metadata_standards

[ISBN]

김선태, 박형선, 이상환, 이상호, 이승복, 이정훈, 최명석, 곽남영

관리 및 유통을 위한 메타데이터 표준 가이드

2015년 11월 일 인쇄

2015년 11월 일 발행

발행처



대전광역시 유성구 대학로 245

☎ 34141

전화 : 042-869-1004

등록 : 1991년 2월 12일 제 5-259호

발행인

한 선 화

인쇄처

(주)
